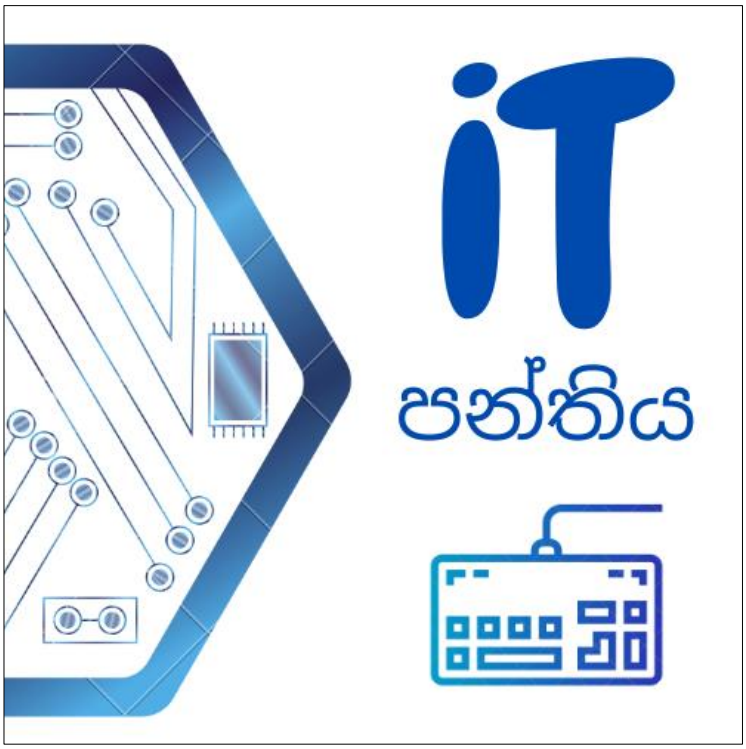
MODEL PAPER 01

Marking Scheme



Model Paper 01 – MCQ Marking Scheme

1) 1	11) 3	21) 2	31) 4	41) N/A
2) 4	12) 1	22) 2	32) 2	42) 1
3) 1	13) 5	23) 3	33) 1	43) 1
4) 1	14) 1	24) 5	34) 1	44) 1
5) 3	15) 2	25) 1	35) 3	45) 3
6) 3	16) 1	26) 4	36) 4	46) 3
7) 2	17) 4	27) 2	37) N/A	47) 2
8) 2	18) 1	28) 5	38) 6	48) 3
9) 1	19) 3	29) 3	39) 4	49) 3
10) 1	20) 1	30) 1	40) 3	50) 4

Q1 – (1) Transistors

Q2 – (4) Formula Translation

Q3 – (1) Electronic Numerical Integrator and Computer

Q4 – (1) A only

The most suitable validation for attendance data is a *presence check* ensuring every student has a recorded entry each day, since attendance must exist for each student daily. Range, format, and data type checks are less relevant because the focus is on confirming data presence, not its format or range.

Q5 – (3) C only

Q6 – (3) A – optical, B – SSD, C – magnetic

Q7 – (2) CPU registers

CPU registers offer the *lowest average access time* because they are the fastest storage locations directly within the processor, much faster than caches, RAM, or SSDs. The CPU retrieves frequently used data from registers almost instantly.

Q8 – (2) (A + B) (A + C)

Q9 – (1) +47

Q10 – (1) 10110010

Q11 – (3) 2 bits and a carry-in

Q12 – (1) 10111101

Q13 – (5) All A, B and C

Q14 – (1) 946

Q15 – (2) 1024 × 8

Q16 – (1) 101101.101

Q17 – (4) All of the mentioned

Q18 – (1) Robert Morris

Q19 – (3) It is an attempt to make a machine or network resource unavailable

A *Denial of Service (DoS)* attack is characterized by overwhelming a server, system, or network with excessive traffic or resource requests so that legitimate users are unable to access it. The goal isn't necessarily to destroy data or corrupt files but to disrupt service availability. Options describing worm infection or data corruption are different types of cyberattacks. The defining feature of DoS is unavailability — hence, describing it as an attempt to make a resource unavailable is the most accurate explanation.

Q20 – (1) We can stop DOS attack completely

This statement is false because it is practically impossible to *completely stop* DoS attacks. Attackers continuously evolve methods to exploit vulnerabilities or overload systems. However, defensive measures like updating operating systems, deploying firewalls, or network-level filtering can *reduce* their impact or frequency. DoS attacks can also persist for extended durations, depending on attacker strength and network defenses. Thus, while partial prevention is possible, total immunity is not achievable, making option (1) the "not true" statement.

Q21 – (2) $F = 0$

Q22 – (2) f = (a + c) (b' + c')

Q23 – (3) Y = AB' + BC + A'C

Q24 – (5) NOR gate

Q25 – (1) SUM = A ⊕ B ⊕ Cin

Q26 – (4) d

Q27 – (2) Multilevel Page Table

In virtual memory systems, storing a single large page table for every process consumes enormous memory. A *multilevel page table* addresses this problem by breaking the page table into smaller segments organized hierarchically. Only the required portions of the table are kept in main memory, reducing memory overhead. For example, a two-level or three-level table divides the address space into parts, each pointing to smaller tables. This structure saves memory while maintaining logical mapping between virtual and physical addresses.

Q28 – (5) Physical memory mapping

The *Process Control Block (PCB)* maintains all essential information about a process needed by the operating system, such as its process state, program counter, CPU scheduling info, and open file descriptors. However, it does not store *physical memory mappings* directly. That responsibility lies with the Memory Management Unit (MMU), which handles address translation. The PCB instead stores references to memory

management information but not the actual physical address map. Hence, physical mapping is excluded from the PCB's contents.

Q29 – (3) To save and restore the process

During a *context switch*, the operating system suspends the currently running process and loads another one for execution. The PCB plays a critical role in this by *saving* the state of the active process — including registers, program counter, and variables — and *restoring* them when the process resumes. This allows multitasking systems to run multiple processes efficiently without losing progress. Without the PCB's save-and-restore mechanism, processes would restart from the beginning each time they gained CPU time.

Q30 – (1) Demand paging

Demand paging improves CPU utilization by loading only the pages actually needed for execution, thus reducing initial load times. However, if memory is insufficient and the CPU constantly swaps pages in and out, the system can experience *thrashing*, where it spends more time paging than executing processes. This condition severely degrades performance. Therefore, while demand paging is efficient under normal workloads, excessive overcommitment of memory can make it counterproductive.

Q31 – (4) Process is compacted

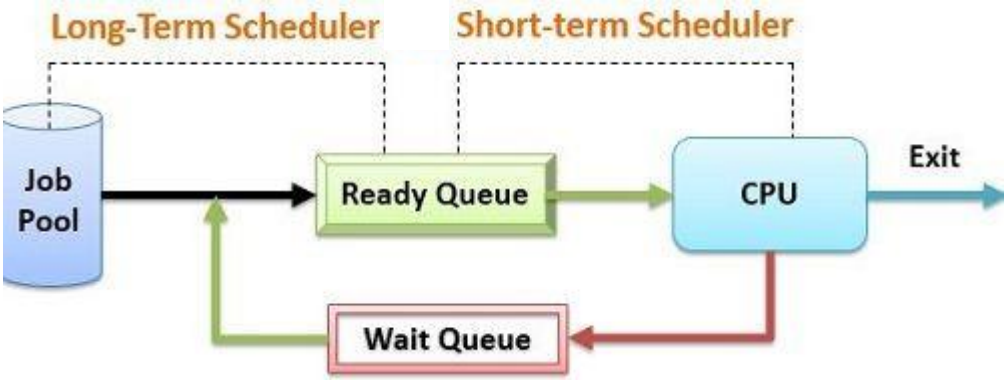
In a swapping system, processes may require contiguous blocks of memory to be reloaded. If such space is unavailable due to fragmentation, *compaction* is used to rearrange existing processes in memory, merging scattered free blocks into one continuous region. This makes it possible to fit the incoming process without terminating or delaying it indefinitely. Compaction is thus a method for managing external fragmentation dynamically in memory systems.

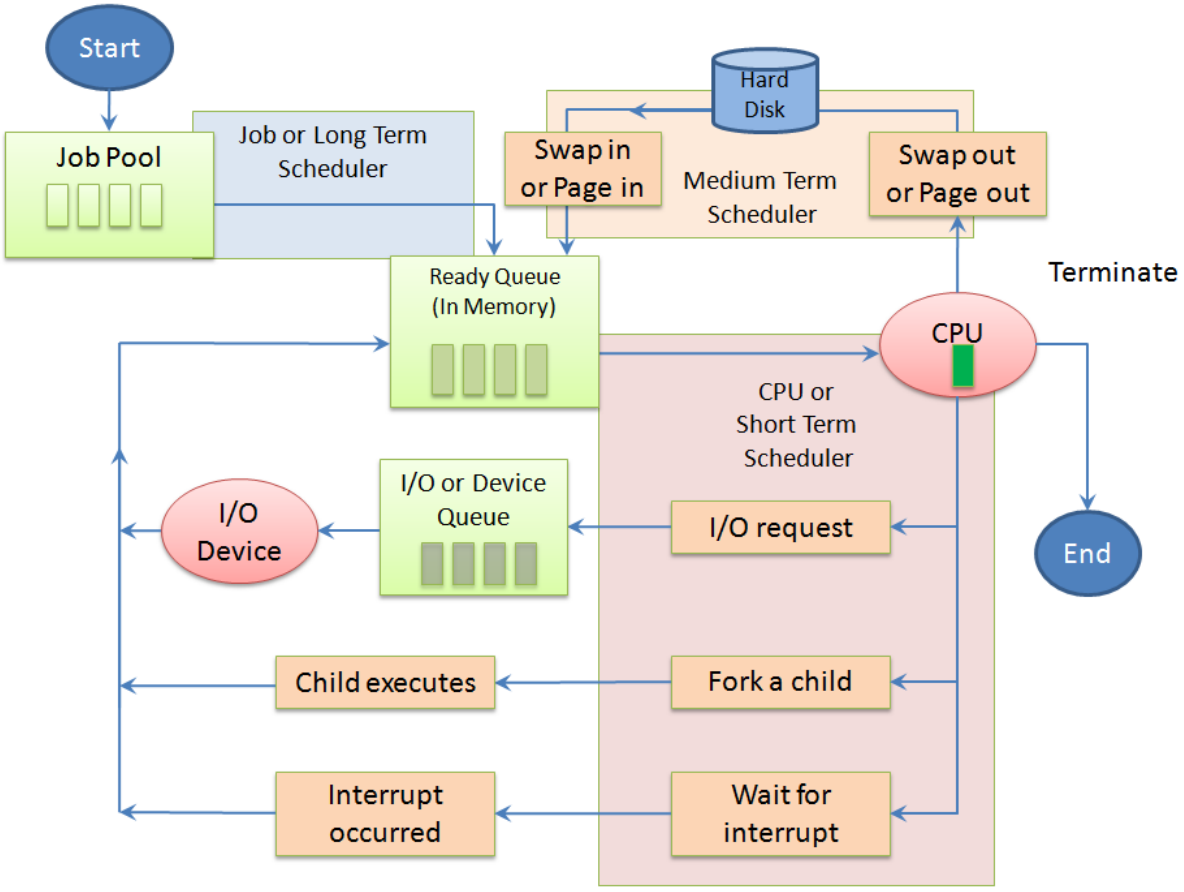
Q32 – (2) The L1 or L2 CPU cache

A program performs best when its *working set* — the most frequently accessed data and instructions — fits entirely within the *CPU cache*, particularly the L1 or L2 levels. These caches are much faster than main memory and drastically reduce access latency. If the working set exceeds cache capacity, the CPU must fetch data from slower memory levels, reducing efficiency. Therefore, optimal performance occurs when the entire working set fits inside these high-speed caches.

Q33 – (1) Long-term scheduler

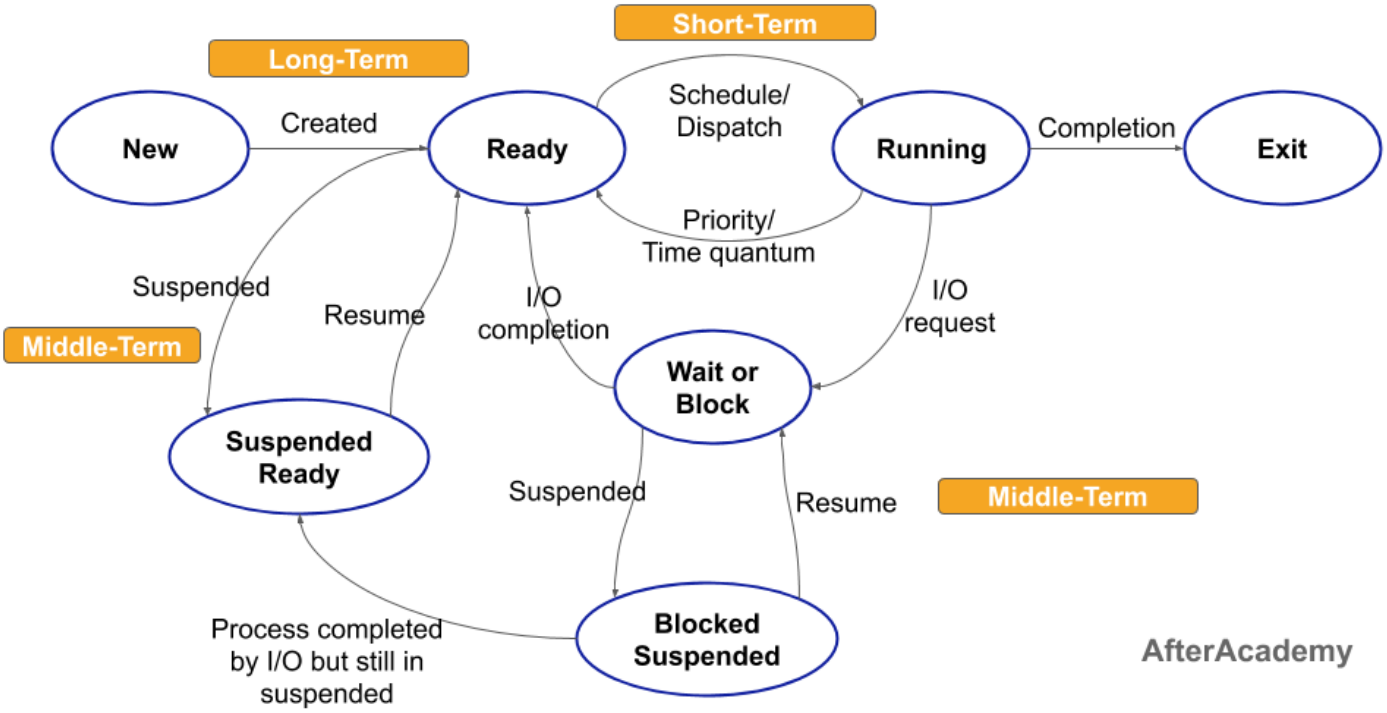
The *long-term scheduler* (also called the admission scheduler) decides which jobs are admitted into the system for processing, thus directly controlling the *degree of multiprogramming*. It balances the load between I/O-bound and CPU-bound processes to ensure efficient utilization of system resources. The short-term scheduler manages CPU assignment among ready processes, but it doesn't determine how many processes enter the system in the first place — that's the long-term scheduler's function.





Q34 – (1) Ready → Running

In the process life cycle, when the CPU scheduler selects a process from the ready queue, the process state transitions from *ready* to *running*. This marks the point where the process actually begins executing instructions on the CPU. Other transitions, such as *running* → *waiting* or *blocked* → *suspended*, occur under different conditions like I/O requests or interruptions. Thus, the "ready to running" transition specifically represents CPU dispatch.



Q35 –

For a **virtual address** VA:

1. **Page number**

$$p = \left\lfloor \frac{VA}{\text{page\_size}} \right\rfloor = \left\lfloor \frac{VA}{8192} \right\rfloor$$

2. **Offset within page**

$$d = VA \bmod \text{page\_size} = VA \bmod 8192$$

3. If page p is present and mapped to **frame** f, then the **physical address**:

$$PA = f \times \text{frame\_size} + d = f \times 8192 + d$$

Q35. Physical address for virtual address 24576

1. Page number and offset:

$$p = \frac{24576}{8192} = 3, \quad d = 24576 \bmod 8192 = 0$$

   So the access is to **page 3, offset 0**.

2. From the page table: page 3 has present bit **0** (and no frame).

So **page 3 is not in memory → page fault**.

There is **no valid physical address** for VA = 24576.

> **Answer (conceptual):** Accessing virtual address 24576 causes a **page fault** (no physical address).

Q36. Which page will cause a page fault if accessed?

A page causes a fault if its present bit = 0.

From the table, present=0 for pages **3, 5, 7**.

Among the options given (0,2,4,6,3), the one that will cause a page fault is:

> **Answer: Page 3.**

Q37. What happens for virtual address 32768?

1. Page and offset:

$$p = \frac{32768}{8192} = 4, \quad d = 32768 \bmod 8192 = 0$$

   So: **page 4, offset 0**.

2. Page 4 in the table: frame = 010_2 (=2), present bit = 1.

   → Page 4 **is in memory**, mapped to **frame 2**.

3. Physical address:

$$PA = 2 \times 8192 + 0 = 16384$$

   In binary (16 bits):

$$PA = 16384_{10} = 0100000000000000_2.$$

So the access is valid and maps to frame **010**.

> **Answer (conceptual):** The address maps to **frame 010** (frame number 2).
> It is **not** a page fault and **not** an invalid access.

Q38. How many frames in physical memory are currently used by this process?

Count the pages with present bit = 1:

- Page 0 → frame 100
- Page 1 → frame 011
- Page 2 → frame 110
- Page 4 → frame 010
- Page 6 → frame 001

That's **5 pages present**, hence **5 frames** used.

> **Answer: 5 frames.**

Q39. What is the frame number mapped to page 2?

From the table: page 2 → frame **110_2**.

> **Answer: 110_2** (which is frame number 6 in decimal).

Q40. If the system needs to swap in Page 5, what will happen?

Page 5 currently has present bit = 0 and no frame listed.

- The first time the process tries to access page 5, this will use the same logic as Q35:
  - Page not present → **page fault**.
- The OS will then handle the fault by choosing/allocating a frame and swapping in page 5.

> **Answer:** A **page fault** will occur (then the OS will allocate a frame and bring page 5 in).

Q41. Physical address for virtual address 16384

1. Page and offset:

$$p = \frac{16384}{8192} = 2, \quad d = 16384 \bmod 8192 = 0$$

So **page 2, offset 0**.

2. Page 2 maps to frame $\mathbf{110_2}$ (= 6), present bit = 1.
3. Physical address:

$$PA = 6 \times 8192 + 0 = 49152$$

In binary (16 bits):

$PA = 49152_{10} = 1100000000000000_2.$

> **Answer (conceptual):** Physical address = $\mathbf{49152_{10}}$ = $\mathbf{1100000000000000_2}$, i.e., frame **110** with offset 0.

Q42 – (1) physical signaling sublayer

The *physical signaling sublayer (PLS)* in the OSI model's physical layer serves as the interface between the *Media Access Control (MAC)* sublayer and the transmission medium. It defines how bits are represented and transmitted as electrical, optical, or radio signals. Essentially, it translates digital data from the MAC into signals suitable for the medium and vice versa, ensuring communication integrity at the hardware level.

Q43 – (1) start and stop signaling

In *asynchronous serial communication*, there is no shared clock between sender and receiver. Therefore, *start and stop bits* are used to mark the beginning and end of each character, helping synchronize data transmission. Flow control is handled at higher layers, not the physical layer. Thus, the physical layer's contribution here lies in ensuring correct start and stop signaling for accurate byte framing during asynchronous transmission.

Q44 – (1) first fit, best fit, worst fit

In dynamic memory allocation, the *first fit* strategy searches memory from the beginning and assigns the first available block large enough to hold the process. This method is generally *faster* than *best fit* or *worst fit*, which require scanning the entire memory list to find the smallest or largest suitable hole, respectively. Although first fit may lead to more fragmentation, its reduced search time makes it quicker in practice.

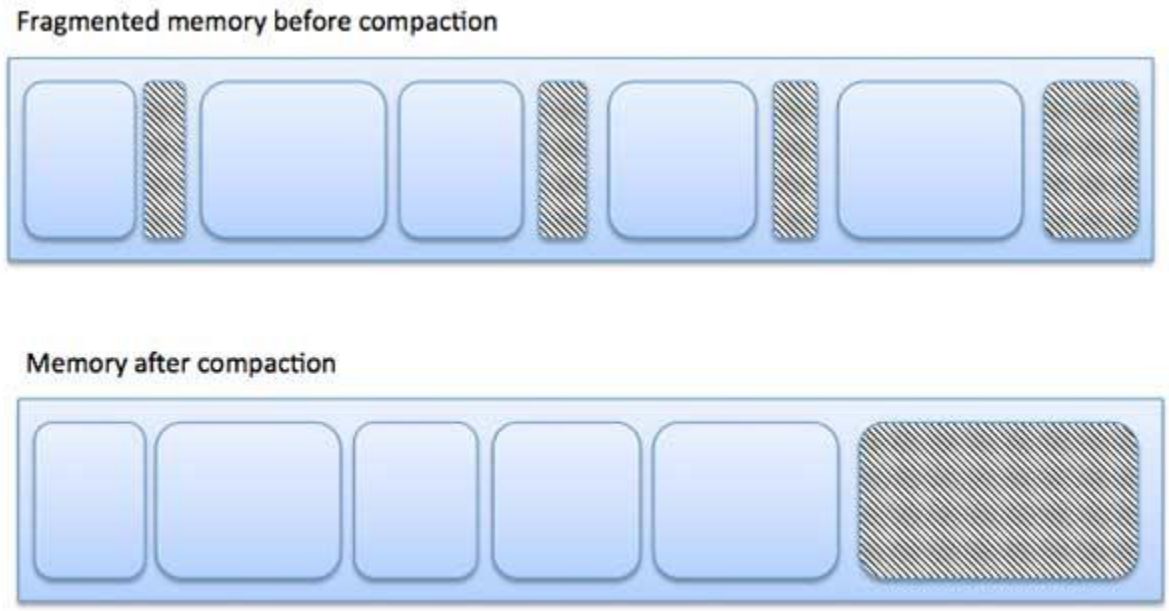
Q45 – (3) very small

When managing free memory spaces (holes), the system must track their locations and sizes. For very small holes, the *bookkeeping overhead*—such as maintaining lists or tables—can sometimes exceed the actual memory available in the hole. In such cases, the overhead of tracking the hole can be *larger than the hole itself*, making it impractical to manage those small fragments efficiently.

Q46 – (3) a technique for overcoming external fragmentation

Compaction is used in systems with variable-sized memory allocation to overcome *external fragmentation*, which occurs when free memory is split into scattered blocks. The system periodically shifts processes in memory to combine these free spaces into a single contiguous block, making it easier to allocate large

processes. This process improves memory utilization and prevents failures due to fragmentation, though it is time-consuming.



Q47 – (2) Data visualization

When computers process and map geographical data, the most critical task is *data visualization*. This involves converting spatial datasets into graphical maps or 3D models, enabling users to interpret geographic patterns, terrain, or resource distributions. Although data storage, collection, and retrieval are essential steps, visualization is the stage where the data is transformed into a meaningful geographic representation — making it the computer's main mapping function.

Q48 – (3) .com

Q49 – (3) Digital Versatile Disc

Q50 – (4) (a), (b) and (c)

Model Paper 01 – Structured Essay

Q1 – (a)

i) → Type of cloud service: Infrastructure as a Service (IaaS)
Explanation: Because you manage the operating system, storage, and software, while Amazon provides the underlying hardware and networking. *(1 mark)*

---

ii) → Type of cloud service: Platform as a Service (PaaS)
Explanation: The platform provides tools and runtime environments for development without managing the underlying infrastructure. *(1 mark)*

---

iii) → Type of cloud service: Software as a Service (SaaS)
Explanation: You access ready-to-use software over the internet without needing to install or manage it.
(1 mark)

(b)

i) → Issue: Phishing attack / Data breach
Explanation: Bob entered his login details on a fake website, allowing attackers to access and steal his documents. *(2 marks)*

---

ii) Suggest two methods to avoid such situations:

1. **Do not click suspicious links** in emails — always verify the sender's address and login directly through the official website.
2. **Enable two-factor authentication (2FA)** to add extra security to cloud accounts. *(2 marks)*

---

iii) Write three advantages of using a cloud service:

1. **Accessibility:** Files and applications can be accessed anytime, anywhere via the internet.
2. **Cost efficiency:** No need to buy or maintain physical hardware.
3. **Scalability and storage:** Easy to increase storage or resources as needed. *(3 marks)*

Q2 – (a)

i) Missing labels

- **A – Input Unit (Input Devices)**
- **B – Output Unit (Output Devices)**
- **C – Main Memory / Memory Unit (RAM)** *(2 marks)*

ii) Examples

- **I (Input devices):** Keyboard, Mouse, Scanner (you could also say Microphone, Webcam)
- **J (Output devices):** Monitor, Printer, Speakers *(2 marks)*

(b) Fill-in-the-blanks

You open Microsoft Word on your computer, it operates based on the **(a) Von Neumann architecture**. Both the program's instructions and your typed text (data) are stored together in the computer's **(b) main memory (RAM)**.
The **(c) control unit** fetches these instructions one by one from memory and sends them to the **(d) arithmetic/logic unit (ALU)** for processing — such as applying text formatting or performing calculations. After processing, the **(e) output unit** displays the results, like the updated document, on the screen.
This process illustrates the **(f) stored-program concept** of the Von Neumann model, where a single memory stores both data and instructions, allowing the CPU to process them efficiently in sequence.

(6 marks)

Q3 – (a)

Scenario 1:

When many computers are connected together and share parts of the same large task.

i) Grid Computing *(1 mark)*

ii) Two advantages:

1. Tasks are completed **much faster** because the workload is shared among many computers.
2. It allows use of **existing computers in different locations**, reducing the need for one expensive supercomputer. *(2 marks)*

iii) Network failure or communication issues between computers can interrupt the process or cause data loss. *(1 mark)*

Scenario 2:

When a computer divides a big job into smaller pieces and processes them at the same time.

iv) Parallel Computing *(1 mark)*

v) Increases processing speed — multiple processors work simultaneously, completing complex tasks like rendering faster. *(1 mark)*

vi) Two drawbacks:

1. **High cost** of multi-processor systems or special hardware.
2. **Programming complexity** — dividing tasks efficiently and combining results can be difficult.

(2 marks)

(b) Fetch–Decode–Execute Cycle

i) Fetch: The **Control Unit** retrieves (fetches) the next instruction from the main memory (RAM).

ii) Decode: The **Control Unit** interprets (decodes) the fetched instruction to understand what action is required.

iii) Execute: The **CPU** carries out (executes) the instruction — e.g., performing a calculation or transferring data. *(2 marks)*

Q4 – (a)

Correct order – [3 1 7 4 5 6 2] *(7 marks)*

• **New: 1**
Amal clicks the game icon on his desktop, and the loading screen appears while the game gets ready to start.

• **Ready: 3**
Rashmi opens a music app and presses play, but the song hasn't started yet because another app is still finishing its work.

• **Running: 4**
Kevin is in an online class using Zoom, talking and listening to others as the app works continuously.

• **Blocked: 2**
Anjana clicks "Print" on her computer and waits as the printer finishes printing the previous page before it continues with hers.

• **Ready Suspended: 6**
Suresh minimizes a photo editor to continue watching a video; the editing app is waiting quietly in the background until he opens it again.

• **Blocked Suspended: 7**
Nimali is uploading family photos to cloud storage when her internet suddenly disconnects. The upload stops and remains paused until the connection returns.

• **Terminated: 5**
Tharindu finishes browsing the internet, closes all tabs, and shuts down the browser completely.

(b)

i) D

ii) E

iii) F

iv) A *(3 marks)*

Model Paper 01 – Essay Type

Q1 – (a)

i) Qualitative and Quantitative Data

- **Qualitative data:** Employee Name, Emp No.

- **Quantitative data:** Basic Salary, OT hours, OT amount, Net Salary. *(1 mark)*

ii) Two information items processed by this sheet
- Total salary paid to all employees.
- Overtime payment calculation for each employee. *(2 marks)*

iii) Two data gathering techniques for HR system
- **Questionnaires or online forms** – to collect employee details.
- **Interviews or observations** – to gather information about attendance and performance. *(2 marks)*

iv) Two advantages of direct data input methods (e.g., fingerprint reader)
- Reduces human error in data entry.
- Saves time and improves accuracy of attendance records. *(2 marks)*

(b)
i) Two suggestions to increase computer performance
- Upgrade RAM or processor.
- Remove unnecessary software and regularly clean temporary files. *(2 marks)*

ii) Two disadvantages of using CRT monitor
- Consumes more electricity.
- Bulky and takes up more space; emits more radiation. *(2 marks)*

iii) Recycle or donate old CRT monitors through certified e-waste recycling programs. *(1 mark)*

iv) The **digital divide** is the gap between individuals or communities that have access to modern information and communication technologies (like the Internet, computers) and those who do not. *(1 mark)*

v) Two ways to mitigate digital divide
- Provide affordable Internet access and digital devices.
- Offer digital literacy training and education programs. *(2 marks)*

Q2 – (a)

i) Two advantages of multicore processors
- Can perform multiple tasks simultaneously (better multitasking and parallel processing).
- Improved overall performance and speed compared to single-core processors. *(2 marks)*

ii) **Cache memory** (or **memory hierarchy** with L1, L2, L3 caches). *(1 mark)*

iii) **Use of cache memory** or **Harvard architecture** (separate data and instruction buses). *(1 mark)*

iv) Three commonly used registers in CPU
- **Accumulator (ACC)**
- **Program Counter (PC)**
- **Instruction Register (IR)**

(Other valid examples: Memory Address Register (MAR), Memory Data Register (MDR)) *(3 marks)*

v) *(2 marks)*

Statement	Answer
1. RAM stores the data and programs that are currently in use.	**TRUE**
2. RAM is used to boot up the computer when power is turned on.	**FALSE**
3. ROM content are retained when power is turned off.	**TRUE**
4. ROM is a volatile memory that can only be read from and not written to.	**FALSE** (ROM is **non-volatile**)

(b)

i) It means that both **program instructions** and **data** are stored in the same memory, and the CPU fetches them as needed using a common bus system. *(1 mark)*

ii) *(3 marks)*

Feature	Primary Memory	Secondary Memory
Definition	Memory directly accessible by CPU (RAM, ROM).	External storage used for long-term data storage (Hard drive, SSD).
Volatility	Usually volatile (e.g., RAM).	Non-volatile.
Speed	Very fast.	Slower.
Usage	Stores data and instructions currently in use.	Stores data permanently.

iii) Is DRAM suitable for memory registers? Explain.

No, DRAM is not suitable for memory registers.

Reason:

- DRAM is slower and needs to be refreshed constantly.
- Registers require **very high-speed memory** for immediate CPU operations.
- **SRAM** (Static RAM) is used instead, as it is faster and does not require refreshing. *(2 marks)*

Q3 –

(a) *(0.5*10 = 5 marks)*

Decimal	Sign & Magnitude	1's complement	2's complement
-115	11110011	10001100	10001101
74	11001010	10110101	10110110
90	01011010	01011010	01011010
-82	11010010	10101101	10101110

(b) i) $01111110_2 = 176_8$ *(1 mark)*

ii) $25.50 = 19.8_{16}$ *(1 mark)*

iii) $62_8 = 110010_2$ *(1 mark)*

iv) Total amount she spent on fruits in decimal = $75.50 = 01001011.10_2$ *(1 mark)*

v) Remaining in decimal = $50.50 = 00110010.10000000_{1\text{'s complement}}$ *(2 marks)*

(c) Logic circuit *(4 marks)*

Q4 –

(a) Logic gate *(4 marks)*

Truth Table *(3 marks)*

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

(b) i) F = AB+BC(B+C)

= AB+BBC+BCC *distributive law*

= AB+BC+BC *identity law*

= AB+BC *idempotent law*

= B(A+C) *distributive law* *(3 marks)*

ii) R = A'BC+AB'C+ABC'+ABC

= C(A'B+AB')+AB(C+C') *distributive law*

=C(A+B)+AB(1) *null law*

=C(A+B)+AB *identity law* *(3 marks)*

(c) 512 GB = 2^9GB

i) $2^9 * 2^{10} = 2^{19}$ MB *(1 mark)*

ii) $2^9 * 2^{30} = 2^{39}$ bytes

$2^{39} * 2^3$ (byte = $8(2^3)$ bits) = 2^{42} bits *(1 mark)*

Q5 – (a)

i) The central processing unit is activated with some **electricity**.

ii) Then the **computer** starts running.

iii) **Power-On Self-Test (POST)** checks the performance of essential hardware components like processor, memory, and storage devices.

iv) The Basic Input-Output System (BIOS) accesses the **CMOS chip** and locates the storage device where the operating system is stored.

v) The Basic Input-Output System (BIOS) copies the operating system files to the **main memory**.

vi) The execution of **operating system** starts and displays the login interface to the user. *(6 marks)*

(b) *(5 marks)*

Column A	**Column B Description**	**Correct Match**
a. Ready	i. The process waits until it gets the processor time	**a → i**
b. Running	f. Currently running process	**b → f**
c. Blocked	j. Processes waiting for input/output	**c → j**
d. Swapped out and blocked	g. Processes transferred to virtual memory, remains in main memory long time for input/output	**d → g**

Column A	Column B Description	Correct Match
e. Swapped out and waiting	h. The processes transferred to virtual memory to free up the main memory for processes higher in priority	**e → h**

(c) i) → Non-preemptive
ii) → Long-term scheduler
iii) → Response time
iv) → Context switching *(4 marks)*

Q6 – (a)

- 17 bits in the virtual address
- 2^{17} bytes = 2^{17} addresses
- Virtual memory size = 2^{17} KB = 128 KB
- Physical memory = 64 KB
- Page (frame) size of physical memory = 4KB = 2^2 bits for the frame no.
- Size of the given program = 64KB = 2^6KB = 2^{16} bytes (which means 16 bits = frame no. + offset)
- No. of addresses in a page = 4096 (0 – 4095) (meaning 2^{12} addresses are in one page)
- So, offset is 12 bits
- Frame no. = 4 bits, Page no. = 5 bits

i) 2^5 pages = 32 pages *(2 marks)*

ii) 2^4 frames = 16 frames *(2 marks)*

iii) no. of bits in the physical address = frame no. (4 bits) + offset (12 bits) *(2 marks)*

iv) 00100 010010110110
00100 = 4 (page no.) = 0011 (frame no.)
Corresponding physical address = 0011 010010110110 *(1 mark)*

v) allow programs larger than physical memory to run *(1 mark)*

(b) i) 2^{32} bits = 2^2 GB = 4 GB *(1 mark)*
ii) 8GB = 2^2 GB * 2^{30} = 2^{32} bytes = 32 bits in a physical address *(1 mark)*

(c) i) **When P1 is preempted so higher-priority P2 runs**

- **P1 (the one that was running):**
  - **PCB state:** *Ready* (moved from Running → Ready due to preemption).
  - **Program Counter (PC):** saved in P1's PCB at the address of the next instruction to execute (the point of interruption).
- **P2 (higher priority, now scheduled):**
  - **PCB state:** *Running* (moved from Ready → Running, or from New → Running if just admitted).
  - **Program Counter (PC):** loaded from P2's PCB (if it was ready earlier) or initialized to its entry point (if starting fresh). *(2 marks)*

ii) **Advantage of keeping a PCB for memory management**

- The PCB holds memory-management info (e.g., base/limit registers, page-table/segment pointers). This lets the OS **protect and isolate address spaces** and **quickly switch/reload mappings** on a context switch (and supports swapping/relocation). *(1 mark)*

iii) **Why does the OS take time during process transitions?**

- Because of **context-switch overhead**: the kernel must save P1's CPU state (registers, PC, flags), update PCBs and queues, possibly change memory maps and flush TLB/caches, load P2's state, and make a scheduling decision. This bookkeeping time is necessary but does no user-level work (pure overhead). *(2 marks)*