

FIRST SEMESTER INTERNAL EXAMINATION

Paper Code: BCA-101 | Fundamentals of Computers & Information Technology (FCIT)

COURSE OUTCOMES (COs) & BLOOM'S TAXONOMY:

CO1: Understand computer architecture, functional units, bus structures & memory hierarchy.

CO3: Evaluate system software, translators (Compilers/Interpreters), & execution pipelines.

CO2: Analyze data representation, error-detection codes, & binary 2's complement arithmetic.

CO4: Analyze OS process management, memory paging, & UNIX multi-user environment.

Time: 2 Hours

Max. Marks: 40

Note: Attempt all questions. Internal choices are indicated under Section B.

Q. No.	Questions (Units 1 & 2: Computer Fundamentals & Operating Systems)	Marks	Unit	CO	BL
SECTION A (Conceptual Analysis & Core Distinctions — Compulsory)					
Q1	Answer all five sub-questions. Each question carries 2 marks: $[5 \times 2 = 10 \text{ Marks}]$				
(a)	Differentiate between: Static RAM (SRAM) and Dynamic RAM (DRAM) with respect to internal cell architecture (flip-flops vs. capacitor-transistor pairs), operational access speed, refresh cycle overhead, and cache integration.	2	1	CO1	L2
(b)	Differentiate between: A <i>Compiler</i> and an <i>Interpreter</i> . Explain how their error diagnosis phase, execution speed, intermediate object code generation, and memory consumption differ during program execution.	2	1	CO3	L2
(c)	Differentiate between: <i>Multiprogramming</i> , <i>Multitasking (Time-Sharing)</i> , and <i>Multiprocessing</i> operating system paradigms in terms of CPU scheduling objectives and concurrent hardware execution.	2	2	CO4	L2
(d)	Differentiate between: <i>Direct Memory Access (DMA)</i> and <i>Interrupt-Driven I/O</i> . Explain why DMA is mandatory for high-bandwidth disk secondary storage operations without monopolizing CPU cycles.	2	1	CO1	L3
(e)	Differentiate between: <i>Hard Links</i> and <i>Soft (Symbolic) Links</i> in the UNIX file system. Explain their behavior when the original target file is unlinked, highlighting inode assignment and cross-filesystem boundaries.	2	2	CO4	L2
SECTION B (Descriptive, Diagram-Based & Analytical Case Problems)					
Q2	(a) Architecture & Diagram: Draw a neat, labeled block diagram of the Von Neumann Architecture showing the interconnection between the Central Processing Unit (ALU, CU, Registers: PC, MAR, MBR, IR, AC) and Main Memory via Address, Data, and Control buses. Explain the sequential three-phase Instruction Cycle (Fetch-Decode-Execute). [5] (b) Data Representation & Arithmetic: Perform binary subtraction using the 8-bit 2's Complement method: Subtract $(43)_{10}$ from $(27)_{10}$. Show full step-by-step bitwise conversions, end-around carry handling, and state clearly whether an arithmetic overflow condition occurred. [5]	10	1	CO1, CO2	L3
OR					
Q2 (Alt)	(a) Memory Hierarchy & Storage Comparison: Draw the hierarchical memory pyramid spanning CPU Registers, L1/L2/L3 Cache, Physical RAM, Solid State Drives (SSD), and Magnetic Hard Disk Drives (HDD). Differentiate between HDDs and SSDs with respect to physical seek latency, rotational delay, and flash wear-leveling endurance. [5] (b) Character Encoding & Error Detection: Differentiate between ASCII, EBCDIC, and Unicode standards. Explain how an Even Parity Bit detects transmission errors, and construct the 8-bit even parity encoded codeword for the ASCII character 'K' (Binary: 1001011). [5]	10	1	CO1, CO2	L3
Q3	(a) Diagram & Process Management: Draw the Five-State Process Transition Diagram (New, Ready, Running, Waiting/Blocked, Terminated) in an operating system. Define the precise triggers for state transitions and contrast the roles of the <i>Short-Term</i>, <i>Medium-Term</i>, and <i>Long-Term Schedulers</i>. [5] (b) Translation Pipeline: Draw and explain the structural compilation pipeline converting C/High-Level source code to an executable binary image: Preprocessor ($\rightarrow .i$), Compiler ($\rightarrow .S$), Assembler ($\rightarrow .O$), Linker, and Loader. Distinguish between <i>Static Linking</i> and <i>Dynamic Linking</i>. [5]	10	2	CO3, CO4	L3
OR					
Q3 (Alt)	Case Scenario: Memory Bottleneck Analysis An analytical workstation with 8 GB physical RAM executes large statistical datasets. When active processes exceed memory capacity, the CPU utilization drops below 8% while disk I/O activity spikes to 100%, causing severe system unresponsiveness. (i) Diagnose this phenomenon. Explain the mechanics of Thrashing in terms of page fault frequency, working set dynamics, and degree of multiprogramming. [3] (ii) Draw an architectural diagram of Virtual Memory Paging illustrating logical-to-physical address translation via the Page Table, and detail the sequence of events during a Page Fault Exception. [4] (iii) Differentiate between <i>Internal Fragmentation</i> and <i>External Fragmentation</i> in memory management. [3]	10	2	CO4	L4