**Name of the Programme: Bachelor of Science in Computer Science (Honours)**

**Course Code: CSC-100**

**Title of the Course: Computer Organization**

**Number of Credits: 4 (3T+1P)**

**Effective from AY: 2023-24**

|  |  |  |
| --- | --- | --- |
| **Pre-requisites for the Course:** | Nil | |
| **Course Objectives:** | 1. Conceptualize the basics of Computer Organizational and Architectural issues and classify the computers based upon performance and machine instructions. 2. Learn various data transfer techniques and the I/O interfaces 3. Estimate and compare performances of various classes of memory 4. Understand the basics of ALU implementation, hardwired and microprogrammed control unit, pipelining and parallel architectures 5. To gain basic practical understanding of shell programming, Assembly language, Linux work environment and commands. | |
| **Theory:** | **Unit I**  Data representation: Data Type Representation, Number System, Signed number, fixed, floating point, character representation, Addition, Subtraction, Multiplication - Shift and Add, Booth’s Algorithm, Division  Introduction to Computer Architecture: Introduction to Computer Architecture, Flynn’s Classification of Computers, Performance Metrics (like Latency, throughput), Fundamental Blocks of Computer (like CPU, I/O subsystems, memory, control unit), computer function, interconnection structures, Bus interconnections | 7 hours  8 hours |
|  | **Unit II**  Memory Hierarchy: Hierarchical memory organization, Types of Memory-internal and external, Cache memory, Memory interleaving.  Peripheral devices: Types of Peripheral Devices, I/O subsystem, programmed I/O, Interrupt-driven I/O, DMA, I/O channels and processors | 7 hours  8 hours |
|  | **Unit III**  Instruction Set Architecture (ISA): Introduction to Instruction Set, Types of ISA; RISC, CISC; Processor Organization, Registers organization, Instruction Execution Cycle, Instruction formats, Addressing Modes; Register Transfer Language (RTL), Assembly Language Programming, X86 -Architecture, ARM Architecture | 15 hours |
| **Practical:** | **Sample Assignments for the Practical Component**   1. Introduction to 8086 architecture and instruction set and Writing assembly language programs in 8086 using MASM or compatible assembler either in windows or Linux 2. Find the sum of 1 + 2 + 3 + n 3. Display the multiplication table of a number 4. Store and retrieve numbers from memory 5. Sort the numbers stored in the memory 6. Installing Linux / Windows Operating System, Partitioning and formatting disk, Installing applications device drivers, working with files, mounting file systems, checking system space, creating, modifying and deleting user accounts 7. Study of Linux Commands 8. Shell Programming in Unix/Linux, arithmetic operations, loops, files   Ex. Write a BASH shell script prime which will accept a number b and display first n prime numbers in standard output.   1. Shell scripting using general-purpose utilities.   Ex. A) Write a menu driven shell script which will print the following menu and execute the given task to display result on standard output.   * 1. Display calendar of current month   2. Display today’s date and time   3. Display usernames those are currently logged in the system   4. Display your name at given x, y position   5. Display your terminal number   6. Exit  1. Shell programming using filters (including grep, egrep, fgrep) | 30 hours |
| **Pedagogy:** | PowerPoint, Tutorials, Hybrid learning | |
| **References/**  **Readings:** | 1. Computer Architecture: A Quantitative Approach by John L. Hennessy & David A. Patterson, 5th Edition, Morgan Kaufmann 2. William Stallings, “Computer Organization and Architecture : Designing for performance”, 9th Edition, Prentice Hall of India. | |
| **Course Outcomes:** | At the end of the course, students will be able to:   1. Recall the basic concepts & terminologies of Computer Organisation. 2. Understand the concepts of data representation, computer & instruction set architecture, memory hierarchy, and peripheral devices. 3. Apply the concepts of data representation, Assembly Language, and performance matrices in solving basic problems. 4. Analyze the multiplication & division algorithms and some design issues in terms of speed, technology, cost, performance, CPU architecture. | |