II Semester

Course 3: Problem Solving using C

Credits -3

Course Objectives

1. To explore basic knowledge on computers
2. Learn how to solve common types of computing problems.
3. Learn to map problems to programming features of C.
4. Learn to write good portable C programs.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand the working of a digital computer and Fundamental constructs of Programming
2. Analyze and develop a solution to a given problem with suitable control structures
3. Apply the derived data types in program solutions
4. Use the 'C' language constructs in the right way
5. Apply the Dynamic Memory Management for effective memory utilization

UNIT-I

Introduction to computer and programming: Introduction, Basic block diagram and functions of various components of computer, Concepts of Hardware and software, Types of software, Compiler and interpreter, Flowcharts and Algorithms

Fundamentals of C: History of C, Features of C, C Tokens-variables and keywords and identifiers, constants and Data types, Rules for constructing variable names, Operators, Structure of C program, Input /output statements in C-Formatted and Unformatted I/O

UNIT-II

Control statements: Decision making statements: if, if else, else if ladder, switch statements. Loop control statements: while loop, for loop and do-while loop. Jump Control statements: break,continue and goto.

UNIT-III

Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation.

Strings: Declaring & Initializing string variables; String handling functions, Character handling functions

UNIT-IV

Functions: Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion, Parameter Passing by address & by value. Local and Global variables. **Storage classes**: automatic, external, static and register.

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic. Pointers and arrays, pointers and functions.

UNIT-V

Dynamic Memory Management: Introduction, Functions-malloc, calloc, realloc, free **Structures:** Basics of structure, structure members, accessing structure members, nested structures, array of

structures, structure and functions, structures and pointers. **Unions** - Union definition; difference between Structures and Unions.

Text Books:

1. E. Balagurusamy, "Programming in ANSI C", Tata McGraw Hill, 6th Edn, ISBN-13: 978-1-25- 90046-2
2. Herbert Schildt, ―Complete Reference with C, Tata McGraw Hill, 4th Edn., ISBN- 13: 9780070411838, 2000
3. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

Reference Books

1. E Balagurusamy, COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
2. Ashok N Kamthane, Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
3. Henry Mullish&Huubert L.Cooper: The Spirit of C An Introduction to modern Programming, Jaico Pub. House,1996.
4. Y kanithkar, let us C BPB, 13 th edition-2013, ISBN:978-8183331630,656 pages.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on computer hardware and software concepts
Evaluation Method: Objective-based quiz assessing knowledge and understanding

Unit 2: Activity: Problem-solving using Decision-Making Statements
Evaluation Method: Correctness of decision-making logic

Unit 3: Activity: Array and String Program Debugging
Evaluation Method: Identification and correction of errors in code

Unit 4: Activity: Pair Programming Exercise on Functions
Evaluation Method: Collaboration and Code Quality

Unit 5: Activity: Structured Programming Assignment
Evaluation Method: Appropriate use of structures and nested structures

II Semester

Course 3: Problem Solving using C

Credits -1

List of Experiments

1. A. Write a program to calculate simple & compound interest
   B. Write a C program to interchange two numbers.
2. Find the biggest of three numbers using C.
3. Write a c program to find the sum of individual digits of a positive integer.
4. A Fibonacci sequence is defined as follows: the first and second terms in the sequenceare 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence.
5. Write a c program to check whether a number is Armstrong or not.
6. Write a c program to generate all the prime numbers between 1 and n, where n is avalue supplied by the user.
7. Write a c program that implements searching of given item in given list
8. Write a c program that uses functions to perform the following: Addition of twomatrices. Multiplication of two matrices.
9. Write a program for concatenation of two strings.
10. Write a program for length of a string with and without String Handling functions
11. Write a program to demonstrate Call by Value and Call by Reference mechanism
12. Write a Program to find GCD of Two numbers using Recursion
13. Write a c program to perform various operations using pointers.
14. Write a c program to read data of 10 employees with a structure of 1.employee id2.aadar no, 3.title, 4.joined date, 5.salary, 6.date of birth, 7.gender, 8.department.
15. Write a Program to demonstrate dynamic arrays using Dynamic Memory Management functions

II Semester

Course 4: Digital Logic Design

Credits -3

Course Objectives

To familiarize with the concepts of designing digital circuits.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Understand how to Convert numbers from one radix to another radix and performarithmetic operations.
2. Simplify Boolean functions using Boolean algebra and k- maps
3. Design adders and subtractors circuits
4. Design combinational logic circuits such as decoders, encoders, multiplexers and demultiplexers.
5. Use flip flops to design registers and counters.

UNIT – I

Number Systems: Binary, octal, decimal, hexadecimal number systems, conversion of numbers from one radix to another radix, r's, (r-1)'s complements, signed binary numbers, addition and subtraction of unsigned and signed numbers, weighted and unweighted codes.

UNIT – II

Logic Gates and Boolean Algebra: NOT, AND, OR, universal gates, X-OR and X-NOR gates, Boolean laws and theorems, complement and dual of a logic function, canonical and standard forms, two level realization of logic functions using universal gates, minimizations of logic functions (POS and SOP) using Boolean theorems, K-map (up to four variables), don't care conditions.

UNIT – III

Combinational Logic Circuits – 1: Design of half adder, full adder, half subtractor, fullsubtractor, ripple adders and subtractors, ripple adder / subtractor.

UNIT – IV

Combinational Logic Circuits – 2: Design of decoders, encoders, priority encoder, multiplexers, demultiplexers, higher order decoders, demultiplexers and multiplexers, realization of Boolean functions using decoders, multiplexers.

UNIT – V

Sequential Logic Circuits: Classification of sequential circuits, latch and flip-flop, RS- latch using NAND and NOR Gates, truth tables, RS, JK, T and D flip-flops, truth and excitation tables, conversion of flip- flops, flip-flops with asynchronous inputs (preset and clear).

Text Books:

1. M. Morris Mano, Michael D Ciletti, “Digital Design”, 5th edition, PEA.

Reference Books

1. Kohavi, Jha, “Switching and Finite Automata Theory”, 3rd edition, Cambridge.
2. 2. Leach, Malvino, Saha, “Digital Principles and Applications”, 7th edition, TMH.
3. 3. Roth, “Fundamentals of Logic Design”, 5th edition, Cengage.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: JAM (Just a Minute) Session: Explaining Radix Conversion
Evaluation Method: Communication Skills and Knowledge Presentation

Unit 2: Activity: Boolean Algebra Assignment
Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Building Adder and Subtractor Circuits
Evaluation Method: Lab Performance and Correctness of Circuit Implementation

Unit 4: Activity: Group Discussion: Applications of Decoders, Encoders, Multiplexers
Evaluation Method: Participation and Critical Thinking

Unit 5: Activity: Quiz on Flip-Flops and Register-Counter Design
Evaluation Method: Quiz Performance and Knowledge Retention

II Semester

Course 4: Digital Logic Design

Credits -1

List of Experiments

The laboratory work can be done by using physical gates and necessary equipment or simulators.

Simulators: https://sourceforge.net/projects/gatesim/ or https://circuitverse.org/ or any free open-source simulator

1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
2. Implementation of the given Boolean functions using logic gates in both SOP and POS forms
3. Realization of basic gates using universal gates.
4. Design and implementation of half and full adder circuits using logic gates.
5. Design and implementation of half and full subtractor circuits using logic gates.
6. Verification of stable tables of RS, JK, T and D flip-flops using NAND gates.
7. Verification of stable tables of RS, JK, T and D flip-flops using NOR gates.
8. Implementation and verification of Decoder and encoder using logic gates.
9. Implementation of 4X1 MUX and DeMUX using logic gates.
10. Implementation of 8X1 MUX using suitable lower order MUX.
11. Implementation of 7-segment decoder circuit.
12. Implementation of 4-bit parallel adder.
13. Design and verification of 4-bit synchronous counter.
14. Design and verification of 4-bit asynchronous counter.