DEBRA THANA SAHID KSHUDIRAM SMRITI MAHAVIDYALAYA

Gangaram Chak, Chak Shyampur, Debra, West Bengal

PROPOSED SYLLABUS (DRAFT) OF

BACHELOR OF SCIENCE WITH COMPUTER SCIENCE (MULTIDISCIPLINARY STUDIES)

3 – YEAR UNDERGRADUATE PROGRAMME

(w.e.f. Academic Year 2024-2025)

Based on

Curriculum & Credit Framework for Undergraduate Programmes (CCFUP), 2023 & NEP, 2020

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
								CA	ESE	TOTAL
B.Sc. in Physical Sc./ Math. & Comp. Sc. with Computer Science	1st	I	**SEMESTER-I**							
			Major (Disc.-A1)	UG/I/COMP/3/ MJ-A1	**T: Introduction to Computers** *(To be studied by the students taken Computer Science as Discipline-A)*	4	3-1-0	15	60	75
			SEC	UG/I/COMP/3/SE-1P	**Problem Solving Using Python**	3	0-0-3	10	40	50
			AEC	UG/I/AEC-1T	Communicative English-1 (***common for all programmes***)	2	2-0-0	10	40	50
			MDC	UG/I/MDC/IT-1T		3	3-0-0	10	40	50
			VAC	UG/I/VAC-1T UG/I/VAC-1P	VAC-01: ENVS (***common for all programmes***)	4	2-0-2	50	50	100
			Minor (Disc.-C1)	UG/I/COMP/3/ MI-C1	**T: Computer Fundamental** **P: Office Automation (Using M.S Office)** *(To be studied by the students taken Computer Science as Discipline-C)*	4	3-0-1	15	60	75
					Semester-I Total	20				400
		II	**SEMESTER-II**							
			Major (Disc.-B1)	UG/II/COMP/3/ MJ-B1	**T: Introduction to Programming in C** **P: Programming in C Lab** *(Same as like A1 for students taken Computer Science as Discipline-B)*	4	3-0-1	15	60	75
			SEC	UG/II/COMP/3/ SE-2P	**Problem Solving Using Python**	3	0-0-3	10	40	50
			AEC	UG/II/AEC-2T	MIL-1 (***common for all programmes***)	2	2-0-0	10	40	50
			MDC	UG/II/MDC/IT-2T	Multi-Disciplinary Course-02 (***to be chosen from the list***)	3	3-0-0	10	40	50
			VAC	UG/II/VAC-2T UG/II/VAC-2P	VAC-02 (***to be chosen from the list***)	4	4-0-0	10	40	50
			Minor (Disc.-C2)	UG/II/COMP/3/ MI-C2	**T: Introduction to Programming in C** **P: Programming in C Lab** *(To be studied by the students taken Computer Science as Discipline-C)*	4	3-0-1	15	60	75
			Summer Intern.	CS	Community Service	4	0-0-4	-	-	50
					Semester-II Total	24				400
					TOTAL of YEAR-1	44	-	-	-	800

MJ= Major Programme (Multidisciplinary), MN = Minor, A/B = Choice of Major Discipline; C= Choice of Minor Discipline; SEC = Skill Enhancement Course, AEC = Ability Enhancement Course, MDC = Multidisciplinary Course, VAC = Value Added Course; CA= Continuous Assessment, ESE= End Semester Examination, T = Theory, P= Practical, L-T-P = Lecture-Tutorial-Practical, MIL = Modern Indian Language, ENVS = Environmental Studie

(Multidisciplinary Studies)

SEMESTER-I

UG/I/COMP/3/MJ-A1: Introduction to Computers Credits 04

Course Objectives:

- Understand the fundamental concepts and characteristics of computers, including their generation and classification.
- Comprehend the basic components of a digital computer, including CPU, ALU, CU, Register set, and memory hierarchy.
- Gain knowledge of communication pathways, input/output devices, and the primary, secondary, cache, and virtual memory.
- Demonstrate proficiency in number systems, including binary, decimal, octal, and hexadecimal, along with arithmetic operations and complement notation.
- Understand data communication principles, components, and modes, as well as the basics of computer networks, network topologies, and types.
- Familiarize themselves with operating systems, their functions, classification, and the concepts of multi-programming, multi-tasking, and multi-processing.
- Gain insights into the booting process and the role of assembler, loader, linker, and interpreter in program execution.

Course Outline:
Introduction to Computers:

1. Introduction: (8 Lectures)

- Definition of computers.
- Classifications of Computes (Micro, Mini, Mainframe, Supercomputers).
- Software/Hardware concepts.
- Terminology (Bit, Byte, Word, Nibble, Computer Languages.

2. Basic Components of Computer: (12 Lectures)

- Computer organization (CPU, CU, ALU, Register set, Communication Pathway, Input/output Devices, Memory Module).
- Understand CPU components: Control Unit (CU), Arithmetic Logic Unit (ALU), and Register set.
- Explore Communication Pathway: Bus, Internal & External Bus, Control, Address & Data Bus.

- Examine Input devices (Keyboard, Pointing devices) and Output devices (Soft copy, hard copy devices).
- Memory Hierarchy: Primary Memory, Secondary Memory, Cache Memory, Virtual Memory.

3. Number System: (15 Lectures)

- Cover Binary, Decimal, Octal, Hexadecimal systems and interconversion.
- Explore Binary-Decimal-Octal Hexadecimal arithmetic, signed & unsigned numbers.
- Learn Complement notation (r's & (r-1)'s complement), Addition & Subtraction using complement notation.
- Dive into Floating-point representation, Computer codes (Weighted binary, Non-weighted binary, Alphanumeric), BCD addition, Gray to Binary & Binary to Gray conversion.

4. Data Communication and Computer Network: (15 Lectures)

- Define data communication, examine characteristics, and components.
- Explore modes, media (guided & unguided) for data transmission.
- Understand Channel capacity, delve into Computer Network concepts (Network topology, Types of networks).
- Explore network devices (Hub, Repeater, Switch, Bridge, Router, Gateway).
- Gain basic understanding of e-mail, Search engines, Chatting, Internet conferencing, and Intranet.

5. Operating System: (10 Lectures)

- Define Operating System (OS), understand functions, necessity, classification (CUI & GUI, Single-user, Multi-user).
- Explore concepts: Multi-Programming, Multi-Tasking, Multi-Processing, Booting Process.
- Understand basics of Assembler, Loader, Linker, and Interpreter.

Suggested Readings:

1. **Sinha, P. K., & Sinha, P. (2017). "Computer Fundamentals: Concepts, Systems & Applications." BPB Publications.**
2. **Rajaraman, V. (2017). "Fundamentals of Computers.", PHI Learning.**
3. **Prakash, S. (2019). "Computer Fundamentals and Programming in C." Laxmi Publications.**
4. **Pradhan, S. (2017). ," Computer Fundamentals: Architecture and Organization." Oxford University Press.**
5. **Bharadwaj, A. S. (2017).," Computer Fundamentals and Applications." Wiley India.**
6. **Deo, N. (2017). ,"Fundamentals of Computers.", Dreamtech Press.**
7. **Acharya, S., & Kamath, M. V. (2017). ,"Computer Fundamentals.", Prentice**

SEC (Skill Enhancement Course)

UG/I/COMP/3/SE-1P: Web design using HTML and CSS credits: 03

Course Objective:

- Develop a comprehensive understanding of key web technologies and the client-server architecture.
- Acquire proficiency in HTML and CSS, exploring diverse tags, elements, and the fundamental structure of HTML documents.
- Master the art of effective webpage styling using CSS, including selectors, properties, and layout techniques.
- Grasp the principles of responsive web design, mobile optimization, and media query implementation.
- Explore advanced features in HTML, including HTML5 elements, and delve into advanced CSS concepts and best practices.
- Create interactive forms, implement animations and transitions, and explore pseudo-classes and pseudo-elements using HTML and CSS.
- Recognize and implement web accessibility best practices, utilizing ARIA roles and attributes.
- Introduce and utilize CSS preprocessors.
- Deepen the ability to create advanced responsive layouts with CSS, exploring intricate layout techniques and media query applications.
- Familiarize themselves with popular CSS frameworks like Bootstrap, integrating pre-built components and styles into web projects.

Course Outline:

1. **Introduction to Web Development**
    - Overview of web technologies.
    - Client-server architecture.
    - Introduction to HTML and CSS.
2. **HTML Fundamentals**
    - Program Explore HTML tags and elements.
    - Understand the document structure in HTML.
    - Learn about forms, multimedia, and semantic HTML.
3. **CSS Styling Techniques**
    - Learn CSS for styling web pages.
    - Explore CSS selectors and properties.
    - Understand layout techniques.

4. **Advanced HTML and CSS**
    - Explore advanced HTML features.
    - Learn about HTML5 and its new elements.
    - Dive into advanced CSS concepts and best practices.
5. **Interactive Elements with HTML and CSS**: **(5 Hours)**
    - Create interactive forms using HTML.
    - Implement animations and transitions with CSS.

HTML and CSS Practical:

- Create a simple webpage with headings, paragraphs, and a list.
- Design a form with various input types (text, password, radio buttons, and checkboxes).
- Build a table displaying information with proper headers and rows.
- Implement an ordered and unordered list to showcase a set of items.
- Develop a webpage with hyperlinks linking to different sections within the same page.
- Design a responsive navigation bar with dropdown menus.
- Create an HTML page that includes multimedia elements such as images, audio, and video.
- Develop a form that utilizes HTML5 semantic elements (e.g., <article>, <section>).
- Construct a simple HTML5 canvas drawing with basic shapes.
- Implement a webpage with an embedded Google Map.
- Style a webpage using internal CSS to change fonts, colors, and background.
- Create a CSS file and link it to an HTML file for external styling.
- Design a responsive layout using Flexbox for better alignment.
- Use CSS Grid to create a two-dimensional layout with rows and columns.
- Implement CSS transitions for smooth effects on hover or click events.
- Style a form with CSS to enhance its visual appeal.
- Customize the appearance of hyperlinks with different states (normal, hover, visited).
- Create a CSS animation for a specific element on your webpage.
- Style a navigation bar to have a fixed position when scrolling.
- Develop a webpage that replicates a login/signup form with proper validation.
- Create a responsive landing page with a hero section and call-to-action buttons.

UG/I/COMP/3/MI-C1: Computer Fundamental (45 Hours) Credits 04

C1-T: Computer Fundamental (T)

Course Outline:

1. **Introduction (3 Hours)**
    - Define computer and discuss its characteristics.
    - Explore the generations of computers and their classifications (Micro, Mini, Mainframe, Super).
    - Examine applications of computers and introduce basic concepts of software and hardware.
    - Cover fundamental notions such as Bit, Byte, Word, Nibble, and various computer languages.
2. : **Basic Components of Computer (7 Hours)**
    - Discuss the basic organization of a digital computer, including CPU, CU, ALU, Register set, and Communication Pathway.
    - Provide a basic explanation of CPU components, such as CU, ALU, and Register set.
    - Define Communication Pathway, covering aspects like Bus, Internal and External Bus, Control, Address, and Data Bus.
    - Explore input and output devices, including keyboards, pointing devices, and Memory hierarchy.

3. **Number System (10 Hours)**
    - Define positional and non-positional number systems, including Binary, Decimal, Octal, and Hexadecimal.
    - Explore binary-decimal-octal-hexadecimal arithmetic, signed and unsigned numbers, and complement notation.
    - Cover addition and subtraction operations using complement notation and floating-point representation of numbers.
    - Discuss computer codes, including weighted binary codes, non-weighted binary codes, and alphanumeric codes.

4. **Data Communication and Computer Network (10 Hours)**
    - Define data communication, its characteristics, components, and modes.
    - Explore data communication media (guided and unguided) and discuss channel capacity.
    - Introduce computer networks, covering network topology, types (LAN, MAN, WAN, CCAN, PAN), and network devices.
    - Provide a basic understanding of email, search engines, chatting, internet conferencing, and intranet

5. **Introduction to System Software and Operating System (15 Hours)**
    - Define the operating system, its functions, and the need for OS.
    - Classify OS based on CUI & GUI and Single or Multi-User systems.
    - Introduce concepts of Multi Programming, Multi-Tasking, and Multi Processing.
    - Explain the booting process and provide a basic understanding of Assembler, Loader, Linker, and Interpreter

UG/I/COMP/3/MI-C1P: Office Automation **Credits 01**

1. **M.S Word**
    - Introduction to Microsoft Word: Comprehensive overview of the Microsoft Word interface, encompassing document creation and seamless navigation.
    - Document Formatting: Mastery of text formatting, paragraph structuring, and meticulous document layout customization.
    - Utilizing Styles and Themes: Proficient application and tailored customization of styles, harnessing document themes for harmonized visual presentation.
    - Effective Document Management: Seamless integration of headers, footers, page numbering, tables, and graphics for comprehensive document management.

2. **Microsoft PowerPoint**
    - Creating Dynamic Presentations: Insightful introduction to the PowerPoint interface, empowering students to craft engaging slides and incorporate compelling content.
    - Customizing Slide Formats: Proficient application of slide layouts, themes, and meticulous customization of slide backgrounds to achieve visual impact.
    - Integrating Multimedia Elements: Seamless insertion of images, audio, and video files to enhance multimedia-rich presentations.
    - Elevating Presentation Delivery: Skillful application of animations and transitions to elevate presentation delivery and captivate the audience.

3. **Microsoft Excel**
    - Essential Spreadsheet Fundamentals: Comprehensive introduction to the Excel interface, encompassing efficient data entry and fundamental formula application.
    - Harnessing Data Analysis Tools: Mastery of sorting, filtering, and conditional formatting tools to facilitate data analysis and interpretation.
    - Visualizing Data with Charts and Graphs: Proficient creation and tailored customization of charts and graphs to effectively visualize data trends.
    - Exploring Advanced Functions: Insightful introduction to advanced functions such as VLOOKUP, SUMIF, and COUNTIF to unlock powerful data manipulation capabilities.

4. **Internet and Email:**
    - Demonstrate how to sign up and sign in to Gmail and navigate through the Gmail interface.
    - Compose an email to someone and include the subject, recipient's email address, and a message describing the event details.
    - Create a meeting in Google Meet and participate in the meeting by interacting with other participants.
    - Join a Google Classroom using the provided class code and access the course materials shared by the instructor.
    - Submit an assignment through Google Classroom and verify the submission.
    - Create a new document in Google Docs and share the document with your classmate and make edits to the document simultaneously with your classmate and observe real-time changes.
    - Create a Google Form for conducting a survey about favorite movie genres and include at least five questions related to movie preferences, share the form link with your classmates and collect responses.

SEMESTER-II

UG/II/COMP/3/MJ-B1: Introduction to Programming in C Credit: 04

Course Objectives:

- Cultivate a profound understanding of programming languages, focusing on 'C,' and encompassing fundamental programming concepts.
- Illuminate key aspects such as Loops, Data reading, stepwise refinement, Functions, Control structures, and Arrays, fostering a holistic understanding of programming fundamentals.
- Develop the capability to analyze real-world problems and adeptly devise solutions through programming in 'C.'
- Prioritize problem-solving skills, emphasizing the creation of effective algorithms applicable in 'C.'
- Equip students with the ability to formulate efficient algorithms for diverse problem scenarios in the 'C' programming language.
- Familiarize students with the various constructs of programming languages, encompassing conditional statements, iteration, and recursion in 'C.'
- Enable students to implement devised algorithms effectively using the "C" language.
- Provide hands-on experience in utilizing fundamental data structures like arrays, stacks, and linked lists for problem-solving in 'C.'
- Empower students with the skills to manage file operations adeptly within the context of "C" programming.

B1-T: Introduction to Programming in C Credits: 03

Course Outline:

1. **Introduction to Programming (4 Hours)**
    - Introduction to various programming styles like procedural, object-oriented, and functional.
    - Understanding the unique features and practical applications of each style.
    - Examining the role of programming in addressing real-world challenges.
    - Introduction to Symbolic Constant, Pre-Processor Directives and Header Files.
    - Understanding the process of analyzing and deconstructing problems, including identifying inputs and output.

- Basics of thinking algorithmically and expressing ideas using pseudocode and Flowchart.
- Introduction to essential algorithms such as searching, sorting, and recursion.

2. **Conditional Statements and Loops (8 Hours)**
    - Overview of the syntax and structure of the C programming language.
    - Understanding different types of variables, data types, and operators in C.
    - Learning basic input and output operations in C.
    - Exploring control flow statements like if-else, nested if-else, switch-case, break, continue and goto.
    - Learning about various loop structures like for loop, while loop, and do-while loop.
    - Applying conditional statements and loops in practical scenarios with examples.
3. **Arrays and Strings (10 Hours)**
    - Introduction to arrays and their manipulation techniques in C.
    - Exploring string handling functions like strcpy, strcat, strlen, etc.
    - Practical exercises focusing on array manipulation and string handling.
4. **Functions and Modular Programming (8 Hours)**
    - Basics of functions including declaration, definition, and invocation.
    - Understanding parameter passing mechanisms such as by value and call by references, Recursive functions, Macro function.
    - Exploring the principles and advantages of modular programming.
    - Implementing modular programs through practice sessions.
5. **Storage Classes and Scope (4 Hours)**
    - Exploring variable scope concepts including local, global, and static.
    - Overview of storage classes such as auto, extern, static, and register.

6. **Structures and Union (6 Hours)**
    - Basics of structure declaration and initialization.
    - Understanding pointer fundamentals including declaration, dereferencing, and arithmetic operations.
    - Exploring the applications of structures and pointers in programming.

7. **Pointers (5 Hours)**
    - Discussion on memory address operators, explanation of declaring pointer types.
    - Assignment and initialization of pointers, performing arithmetic operations using pointers.

- Functions that utilize pointers.
- Exploring the connection between arrays and pointers
- Handling pointers within structures.

B1-P: Programming in C (C Lab) **Credits: 01**

- Write a C program to check whether a given number is even or odd.
- Write a C program to calculate the area of a rectangle.
- Write a C program to calculate the sum of first n even numbers.
- Write a C program to read a number and find factorial.
- Write a C program to find largest number among three numbers.
- Write a C program to check whether a given number is Amstrong or not.
- Write a program to check whether a given number is prime or not.
- Write a program to generate the Fibonacci series up to a given number.
- Write a program to check whether a given string is a palindrome or not.
- Write a C program to implement a simple calculator that performs addition, subtraction, multiplication, and division.
- Implement a C program to find the largest element in an array.
- Write a C program to concatenate two strings.
- Write a C program to swap two numbers.
- Write a C program to perform linear search to find the position of a given element in an array.
- Write a C program to sort an array of integers in ascending order.

Reference Books:

For C Programming Language

- Byron S Gottfried “Programming with C”, Fourth edition, Tata McGrawhill.
- E. Balagurusamy, “Programming with ANSI-C”, Eight Edition,2008, Tata McGraw Hill.
- Kanetkar Y, “Let us C”, BPB Publications, 2017.
- Brian W. Kernighan & Dennis Ritchie, “The C Programming Language”, Second Edition, Pearson.

SEC (Skill Enhancement Course)

UG/II/COMP/3/SE-2P: Problem Solving Using Python Credit: 03

Course Objective:

- Introduce fundamental features of Python programming, focusing on industry standards.
- Enable students to apply advanced Python programming features to solve real-world problems.
- Instill a solid understanding of Python programming concepts, libraries.

Course Outline:

- Develop the capability to create applications in various fields using Python.
- Explore computer systems and the Python programming language.
- Emphasize computational thinking and cover Python data types, expressions, operators, variables, assignments, strings, lists, and the Python standard library.
- Dive into imperative programming, covering Python modules and built-in functions like print() and eval().
- Develop user-defined functions and assignments, emphasizing parameter passing.
- Focus on text data, files, and exceptions, covering strings, formatted output, files, errors, and exceptions.
- Introduce execution control structures, decision control, and the IF statement.
- Cover For LOOP and iteration patterns, including two-dimensional lists, while loop, additional loop patterns, and iteration control statements.
- String, List, Tuples, Dictionaries, Sets.

Python Practical:

- Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users' choice.
- Check if a number is prime or not.

- Write a Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:

  Grade A: Percentage >=80
  Grade B: Percentage>=70 and <80
  Grade C: Percentage>=60 and <70
  Grade D: Percentage>=40 and <60
  Grade E: Percentage<40

- Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
- Write a Program to display the first n terms of Fibonacci series.
- Write a Program to find factorial of the given number.
- Write a Program to find sum of the following series for n terms: 1 – 2/2! + 3/3!--------- n/n!
- Write a Program to calculate the sum and product of two compatible matrices.
- Create two matrices and perform matrix multiplication using NumPy.
- Write a program to find the square root of a given number.

REFERENCE BOOKS:

1. "Python Programming: A Modular Approach with Graphics, Database, Mobile, and Web Applications" by Sheetal Taneja & Naveen Kumar, Pearson, 2017.
2. "Python: The Complete Reference" by Martin C. Brown, Osborne/McHraw Hill, 2001.
3. "Core Python Programming" by Wesley J. Chun, Pearson Education, Second Edition, 2007.
4. Introduction to Computing Using Python: An Application Development Focus" by Ljubomir Perkovic, John Wiley & Sons, 2012.

UG/II/COMP/3/MI-C2: Introduction to Programming in C Credit: 04

Course Objectives:

- Cultivate a profound understanding of programming languages, focusing on 'C,' and encompassing fundamental programming concepts.
- Illuminate key aspects such as Loops, Data reading, stepwise refinement, Functions, Control structures, and Arrays, fostering a holistic understanding of programming fundamentals.
- Develop the capability to analyze real-world problems and adeptly devise solutions through programming in 'C.'
- Prioritize problem-solving skills, emphasizing the creation of effective algorithms applicable in 'C.'
- Equip students with the ability to formulate efficient algorithms for diverse problem scenarios in the 'C' programming language.
- Familiarize students with the various constructs of programming languages, encompassing conditional statements, iteration, and recursion in 'C.'
- Enable students to implement devised algorithms effectively using the "C" language.
- Provide hands-on experience in utilizing fundamental data structures like arrays, stacks, and linked lists for problem-solving in 'C.'
- Empower students with the skills to manage file operations adeptly within the context of "C" programming.

C2-T: Introduction to Programming in C **Credits: 03**

Course Outline:

1. **Introduction to Programming (4 Hours)**
    - Introduction to various programming styles like procedural, object-oriented, and functional.
    - Understanding the unique features and practical applications of each style.
    - Examining the role of programming in addressing real-world challenges.
    - Introduction to Symbolic Constant, Pre-Processor Directives and Header Files.
    - Understanding the process of analyzing and deconstructing problems, including identifying inputs and output.

- Basics of thinking algorithmically and expressing ideas using pseudocode and Flowchart.
- Introduction to essential algorithms such as searching, sorting, and recursion.

2. **Conditional Statements and Loops (8 Hours)**
    - Overview of the syntax and structure of the C programming language.
    - Understanding different types of variables, data types, and operators in C.
    - Learning basic input and output operations in C.
    - Exploring control flow statements like if-else, nested if-else, switch-case, break, continue and goto.
    - Learning about various loop structures like for loop, while loop, and do-while loop.
    - Applying conditional statements and loops in practical scenarios with examples.
3. **Arrays and Strings (10 Hours)**
    - Introduction to arrays and their manipulation techniques in C.
    - Exploring string handling functions like strcpy, strcat, strlen, etc.
    - Practical exercises focusing on array manipulation and string handling.
4. **Functions and Modular Programming (8 Hours)**
    - Basics of functions including declaration, definition, and invocation.
    - Understanding parameter passing mechanisms such as by value and call by references, Recursive functions, Macro function.
    - Exploring the principles and advantages of modular programming.
    - Implementing modular programs through practice sessions.
5. **Storage Classes and Scope (4 Hours)**
    - Exploring variable scope concepts including local, global, and static.
    - Overview of storage classes such as auto, extern, static, and register.

6. **Structures and Union (6 Hours)**
    - Basics of structure declaration and initialization.
    - Understanding pointer fundamentals including declaration, dereferencing, and arithmetic operations.
    - Exploring the applications of structures and pointers in programming.

7. **Pointers (5 Hours)**
    - Discussion on memory address operators, explanation of declaring pointer types.
    - Assignment and initialization of pointers, performing arithmetic operations using pointers.

- Functions that utilize pointers.
- Exploring the connection between arrays and pointers
- Handling pointers within structures.

C2-P: Programming in C (C Lab) **Credits: 01**

- Write a C program to check whether a given number is even or odd.
- Write a C program to calculate the area of a rectangle.
- Write a C program to calculate the sum of first n even numbers.
- Write a C program to read a number and find factorial.
- Write a C program to find largest number among three numbers.
- Write a C program to check whether a given number is Amstrong or not.
- Write a program to check whether a given number is prime or not.
- Write a program to generate the Fibonacci series up to a given number.
- Write a program to check whether a given string is a palindrome or not.
- Write a C program to implement a simple calculator that performs addition, subtraction, multiplication, and division.
- Implement a C program to find the largest element in an array.
- Write a C program to concatenate two strings.
- Write a C program to swap two numbers.
- Write a C program to perform linear search to find the position of a given element in an array.
- Write a C program to sort an array of integers in ascending order.

Reference Books:

For C Programming Language

- Byron S Gottfried “Programming with C”, Fourth edition, Tata McGrawhill.
- E. Balagurusamy, “Programming with ANSI-C”, Eight Edition,2008, Tata McGraw Hill.
- Kanetkar Y, “Let us C”, BPB Publications, 2017.
- Brian W. Kernighan & Dennis Ritchie, “The C Programming Language”, Second Edition, Pearson.