# CENTRE FOR DISTANCE AND ONLINE EDUCATION
**BHARATHIDASAN UNIVERSITY, TIRUCHIRAPPALLI - 620 024.**

## B.Sc. (Information Technology) -CHOICE BASED CREDIT SYSTEM – LEARNINIG OUTCOMES BASED CURRICULUM FRAMEWORK (CBCS - LOCF)

**(Applicable to the candidates admitted from the academic year 2026-2027 onwards)**

| Sem. | Part | Course | Title | Credit | Exam Hours | Marks | | Total |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Int. | Ext. | |
| I | I | Language Course – I (Tamil*/Other Languages +#) | | 3 | 3 | 25 | 75 | 100 |
| | II | English Course - I | | 3 | 3 | 25 | 75 | 100 |
| | III | Core Course – I (CC) | Programming in C and Data Structures | 5 | 3 | 25 | 75 | 100 |
| | | Core Practical – I(CP) | Programming in C Lab | 4 | 3 | 40 | 60 | 100 |
| | | First Allied Course – I (AC) | Algebra and calculus | 4 | 3 | 25 | 75 | 100 |
| | | First Allied Course – II(AC) | | -- | -- | -- | -- | -- |
| | IV | Value Education | | 2 | 3 | 25 | 75 | 100 |
| | | | **TOTAL** | **21** | **-** | **-** | **-** | **600** |
| II | I | Language Course - II (Tamil*/Other Languages +#) | | 3 | 3 | 25 | 75 | 100 |
| | II | English Course - II | | 3 | 3 | 25 | 75 | 100 |
| | III | Core Course – II(CC) | Programming in Java | 5 | 3 | 25 | 75 | 100 |
| | | Core Practical - II(CP I-III) | Programming in Java Lab | 4 | 3 | 40 | 60 | 100 |
| | | First Allied Course – II (AC) | Numerical analysis and probability | 2 | 3 | 25 | 75 | 100 |
| | | First Allied Course – III (AC) | Operations research | 4 | 3 | 40 | 60 | 100 |
| | IV | Environmental Studies | | 2 | 3 | 25 | 75 | 100 |
| | | | **TOTAL** | **23** | **-** | **-** | **-** | **700** |

## List of Allied Courses

**First Allied Course** — Mathematics

**Second Allied Course** — Applied Physics

| Sl. No. | Part | Types of the Course | No. of Courses | No. of Credits | Marks |
|---|---|---|---|---|---|
| 1. | I | Language Courses | 4 | 12 | 400 |
| 2. | II | English Courses | 4 | 12 | 400 |
| 3. | III | Core Courses | 9 | 45 | 900 |
| 4. | | Core Practical | 6 | 24 | 600 |
| 5. | | Allied Courses | 4 | 16 | 400 |
| 6. | | Allied Practical | 2 | 4 | 200 |
| 7. | | Major Based Elective Courses | 2 | 8 | 200 |
| 8. | | Project | 1 | 7 | 100 |
| 9. | IV | Value Education | 1 | 2 | 100 |
| 10. | | Environmental Science | 1 | 2 | 100 |
| 11. | | Soft Skills Development | 1 | 2 | 100 |
| 12. | | Non-Major Elective | 2 | 4 | 200 |
| 13. | V | Gender Studies | 1 | 2 | 100 |
| | **Total** | | **38** | **140** | **3800** |

**PROGRAMME OUTCOMES:**

- Graduates will be able to comprehend the basic concepts learnt and apply in real life situations with analytical skills.
- Graduates with acquired skills and enhanced knowledge will be employable/ become entrepreneurs or will pursue higher Education.
- Graduates with acquired knowledge of modern software tools will be able to contribute effectively as software engineers.
- Graduates will be able to comprehend the related concepts to Computer Science with Allied papers
- Graduates will be imbibed with ethical values and social concerns to ensure peaceful society.

## PROGRAMME SPECIFIC OUTCOMES:

- After completing the B.Sc.Information Technology Programme,the graduates would have Acquired the required knowledge in the Hardware and Software aspects of Computer Science domain and the art of programming.
- Understood the development methodologies of software systems and the ability to analyze design and develop computer applications for real life problems.
- Gained knowledge and skills to collaborate and communicate with peers in IT/ITES industries
- The ability to understand, adjust and adapt with the dynamic technical environment for the growth of IT industry.
- The capability to transfer, with the skills gained, to provide innovative and novel solutions by maintaining ethical norms for the betterment of human society.

**First Year** **CORE COURSE I** **Semester I**

# PROGRAMMING IN C AND DATA STRUCTURES
## (Theory)

## COURSEOBJECTIVES:

- To know about the basics of C programming, control and looping structures, and programming with it.
- To understand arrays, pointers, and string processing in C language.
- To know about the basic concepts in data structures.

### UNIT – I

**Basics of C:** History of C and its importance – Structure of a C program – Data Types – Constants and Variables – Operators and Expressions – Order of Precedence – Evaluation of Arithmetic Expressions – Type Conversion – Decision Statements: if, if-else, and nested if statements.

### UNIT – II

**Loop Structures:** For loop, while, do-while loop – **Arrays:** One-dimensional array, two-dimensional arrays, character arrays and strings – **Functions:** Function with arrays – Function with decision and looping statements – Recursion.

### UNIT – III

**Pointers:** Introduction – Pointer expressions – Chain of pointers – Pointers and arrays – Array of pointers – Pointers as function arguments – Functions returning pointers – Pointers to functions – Function pointer – **Structures:** Declaration, initialization, array of structures – Pointer to structures – Structures and functions – Typedef – Enumerated data types – Unions.

### UNIT – IV

**String Processing:** Standard string library functions – **Files:** Introduction and file functions – Writing and reading in text mode – Simple applications: Display the contents of a file – Write data to a file – Append data to an existing file – File I/O – Reading and writing structures.

### UNIT – V

**Stack:** LIFO concept – Stack operations – Array implementation of stack – **Queue:** FIFO concept – Queue operations – Array implementation of queue – **Singly Linked List:** Concepts – Operations – **Doubly Linked List:** Concepts – Operations – **Trees:** General trees – Binary trees.

### UNIT – VI: CURRENT CONTOURS

*(For Continuous Internal Assessment Only)*

Contemporary developments related to the course during the semester concerned.

### REFERENCES

1. E. Balagurusamy, *Programming in ANSI C,* Tata McGraw Hill, New Delhi, Seventh Edition, 2016.

2. E. Horowitz, S. Sahni and Susan Anderson Freed, *Fundamental Data Structures in C*, 2nd ed., Orient BlackSwan Publisher, 2009.
3. Byron S. Gottfried, *Programming with C*, Schaum's Outline Series, Tata McGraw Hill Edition, New Delhi, 1991.
4. E. Karthikeyan, *A Textbook on C Fundamentals, Data Structures and Problem Solving*, Prentice-Hall of India Private Limited, New Delhi, 2008.
5. Yashavant Kanetkar, *Let Us C*, BPB Publications, Tenth Edition, New Delhi, 2010.
6. Szuhay, Jeff, *Learn C Programming: A Beginner's Guide to Learning C Programming the Easy and Disciplined Way*, Packt Publishing, 2020.
7. Jena, Sisir Kumar, *C Programming: Learn to Code*, CRC Press, 2021.
8. https://www.tutorialspoint.com/cprogramming/index.htm
9. https://www.w3schools.in/data-structures/intro

## COURSE OUTCOMES

**Upon successful completion of this course, the students would be able to:**

- Summarize the basic knowledge required to develop C programs.
- Manipulate looping structures, arrays, and functions.
- Apply and write programs for solving real-world problems.
- Create, open, read, manipulate, write, and close files.
- Understand the basic concepts of data structures.

**First Year** **CORE PRACTICAL I** **Semester I**

**PROGRAMMING IN C LAB**

**(Practical)**

## COURSE OBJECTIVES:

- To understand the programming fundamentals of C language.
- To impart writing skills in C programming and data structures for a list of problems.
- To impart hands-on training for writing C programs using computers.

## PROGRAMMING EXERCISES

1. **Write a program:**

   (i) To convert temperature from degree Centigrade to Fahrenheit.

   (ii) To find whether a given number is even or odd.

   (iii) To find the greatest of three numbers.

2. **Write a program** to display Monday to Sunday using a switch statement.
3. **Write a program** to display the first ten natural numbers and their sum.
4. **Write a program** to perform multiplication of two matrices.
5. **Write a program:**

   (i) To find the maximum number in an array using a pointer.

   (ii) To reverse a number using a pointer.

   (iii) To add two numbers using pointers.

6. **Write a program** to solve a quadratic equation using functions.
7. **Write a program** to find the factorial of a number using recursion.
8. **Write a program** to demonstrate call by value and call by reference.
9. **Write a program** to create a file containing student details.
10. **Write a program** to implement a stack using a singly linked list and implement a queue using a linked list.

## COURSE OUTCOMES

**Upon successful completion of this course, the students would be able to:**

- Relate the use of language constructs to solve simple programs.
- Develop programs for various concepts in C language.
- Understand and trace the execution of the list of programs.
- Understand the usage of file handling in C programming.
- Solve data problems related to data structures.

*******

**First Year** **CORE COURSE II** **SemesterII**

**PROGRAMMING IN JAVA**

**(Theory)**

**COURSE OBJECTIVES**

- To acquire programming skills with Java.
- To implement object-oriented concepts using the Java language.
- To learn the art of GUI programming with Applets.

**UNIT – I Foundation, Essentials, Control Statements and Classes & Objects:**

Stages of Java – Origin of Java – Challenges – Features – Object-Oriented Programming – **Java Essentials:** Elements – API – Variables – Primitive data types – String class – Operators – Combined assignment operators – Conversion – Scope – Comments – Keyboard input – **Control Statements:** if, if-else, nested if and if-else-if statements – Logical operators – Comparison – Conditional operator – switch – Increment and decrement – while, do-while and for loops – Nested loops – break and continue – **Classes and Objects:** Classes and objects – Modifiers – Passing arguments – Constructors – Package and import – Static class members – Method overloading – Constructor overloading – Returning objects – this variable – Recursion – Nested and inner classes – Abstract classes and methods.

**UNIT – II Arrays, String Handling, Inheritance, Interface and Packages:**

**Introduction – Processing array – Passing arrays – Returning arrays – String arrays – Two-dimensional arrays – Arrays with three or more dimensions –** String Handling: **String class – Concatenation – Comparison – Substring – Methods – Other methods – StringBuffer, StringBuilder and StringTokenizer classes –** Inheritance: **Basics – Inheriting and overriding superclass methods – Calling superclass constructor – Polymorphism – Inherit from different classes – Abstract classes – Final class –** Interfaces: **Basics – Multiple interfaces – Multiple inheritance using interface – Multilevel interface –** Packages: **Create and access packages in NetBeans IDE – Static import and package class – Access specifiers.**

**UNIT – III Exception Handling, I/O and File Handling, and Multithreading:**

**Introduction – try and catch block – Multiple catch block – Nested try – finally block – throw statement – Exception propagation – throws clause – Custom exception – Built-in exception –** Multithreading: **Introduction – Threads – Thread creation – Life cycle – Joining a thread – Scheduler and priority – Synchronization – Inter-thread communication – Thread control – Thread pool – Thread group – Daemon thread –** Files and I/O Streams: **File class – Streams – Byte streams – Filtered byte streams – Random Access File class – Character streams.**

**UNIT –** IV **Applet and GUI – Part I:**

**Fundamentals – Applet class – Life cycle – Steps for applet program – Passing values through parameters – Graphics – Event handling –** GUI: **GUI – Creating windows – Dialog boxes – Layout managers – AWT component classes – Swing component classes – Applications of AWT controls.**

**UNIT – V GUI – Part II and Java Database Connectivity:**

**Event handling – AWT components – AWT graphics classes – Swing controls – Applications using Swing and AWT –** Java Database Connectivity: **Types of drivers – JDBC architecture – JDBC classes and interfaces – Steps in JDBC applications – Creating a new database and table with JDBC.**

## UNIT – VI: CURRENT CONTOURS

*(For Continuous Internal Assessment Only)*

Contemporary developments related to the course during the semester concerned.

## REFERENCES


1. S. Sagayaraj, R. Denis, P. Karthik & D. Gajalakshmi, *Constructive Java Programming*, Universities Press, 2021.
2. E. Balagurusamy, *Programming with JAVA*, Tata McGraw Hill, New Delhi, 2019.
3. C. Muthu, *Programming with JAVA*, Vijay Nicole Imprints Private Limited, Chennai, Second Edition, 2011.
4. Bruce Eckel and Chuck Allison, *Thinking in Java*, Prentice Hall Publications, 2006.
5. Malina Pronto, *Java: How to Learn Java Programming: How to Improve Your Java Coding in 2020/2021: 5 Programming Languages to Learn for Beginners in Tech*, Independently Published, 2020.
6. Nick Samoylov, *Learn Java 12 Programming: A Step-by-Step Guide to Learning Essential Concepts in Java*, Packt Publishing, 2019.
7. https://www.javatpoint.com/java-tutorial


## COURSE OUTCOMES

**Upon successful completion of this course, the students would be able to:**

- Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, encapsulation, and method overloading.
- Identify members of a class and implement them.
- Create Java application programs using sound OOP practices (e.g., interfaces and APIs) and proper program structuring (e.g., using access control identifiers) and create user-defined packages for specific tasks, incorporating reusability concepts and error/exception handling.
- Develop programs using the Java standard class library.
- Develop software using the Java programming language, using Applets, AWT controls, and JDBC.

*****

**First Year** **CORE PRACTICAL II** **Semester II**

**PROGRAMMING IN JAVA LAB**

**(Practical)**

## COURSE OBJECTIVES:

### COURSE OBJECTIVES

- To understand the basics of Java programs and their execution.
- To learn concepts like inheritance, packages, and interfaces.
- To understand the life cycle of Applets, database connectivity, and their functionality.

### PROGRAMMING EXERCISES

1. **Write a program** to sort the given numbers using arrays.
2. **Write a program** to implement the FIND and REPLACE operations in the given text.
3. **Write a program** to implement a calculator to perform basic arithmetic operations using constructors.
4. **Write a program** to find the student's percentage and grade using command-line arguments.
5. **Write a program** to draw a circle, triangle, or square using polymorphism and inheritance.
6. **Implement multiple inheritance concepts in Java using interfaces.** You can choose your own example of a company, educational institution, or a general concept that requires the use of interfaces to solve a particular problem.
7. **Write a program** to create threads and perform operations like start, stop, suspend, and resume.
8. **Write a program** to develop an Applet to play multiple audio clips using multithreading.
9. **Write a program** to retrieve employee data from a file.
10. **Write a program** to retrieve student data from a database.

### COURSE OUTCOMES

**Upon successful completion of this course, the students would be able to:**

- Develop Java programs to understand OOP concepts.
- Write Java programs for classes and objects.
- Develop simple programs with multiple threads.
- Write Java programs using Applets.
- Develop Java programs to connect databases and files.