

# SHRI MAHAVEER COLLEGE



*Shri Mahaveer College*

Affiliated to the University of Rajasthan

(A Co-educational English Medium PG College)

**PROGRAM OUTCOMES**

**&**

**COURSE OUTCOMES**

**BCA**

**As Per NEP-2020**

**Session 2023-24 (Sem. I & II)**

**Session 2024-25 (Sem. III)**

**(Department of Computer Science)**

## BCA Programme

### **Programme Specific Outcomes (PSOs)**

<b>PSO Number</b>	<b>Upon completion of BCA Degree Programme the graduates will be able to</b>
PSO 1	Develop proficiency in problem solving and logical thinking skill in computer science.
PSO 2	Appraise in-depth expertise and learning the knowledge of programming languages, web designing, networking and Software development cycle.

### **Programme Outcomes (POs)**

<b>PO Number</b>	<b>Upon completion of BCA Degree Programme the graduates will be able to</b>
PO-1	Discipline knowledge: Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity, engineering problems.
PO-2	Problem Solving: Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms of mathematics, natural sciences, and engineering sciences.
PO-3	Difficulty Analysis: Talent to classify, significantly evaluate and prepare complex computing problems using fundamentals of computer knowledge and request domains, appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO-4	Design and Development of Solutions: Ability to design and development of algorithmic solutions to real/world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems .data, and synthesis of the information to provide valid conclusions.

PO-5	Accomplish Investigations of Compound Computing Troubles: Ability to invent and ways experiments interpret data and present well up to date conclusions.
PO-6	Application Systems Knowledge: Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.
PO-7	Modern Tool Usage: Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
PO-8	Mission Administration: Skill to recognize administration and Computing philosophy with computing acquaintance to supervise projects in multidisciplinary environments.
PO-9	Communication: Must have a reasonably good communication knowledge both in oral and writing.
PO-10	Ethics on Professional, Environment and Society: Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems..
PO-11	Motivation to take up Higher Studies: Inspiration to continue educations towards advanced studies on Computer Science.

# COURSE OUTCOMES

## BCA Part I Sem-I

<b>BCA-51T-101: Programming in C</b>	
After completion of course:	
CO1	Explain the basic programming concepts and syntax of the C language.
CO2	Design and implement algorithms to solve simple programming problems.
CO3	Perform input and output operations using programs in C.
CO4	Write programs that perform operations on arrays Course Content.

<b>BCA-51P-102: Programming in C Lab</b>	
After completion of course:	
CO1	Explain the basic programming concepts and syntax of the C language.
CO2	Design and implement algorithms to solve simple programming problems.
CO3	Perform input and output operations using programs in C.
CO4	Write programs that perform operations on arrays Course Content.

<b>BCA-51T-103: Web Application Development</b>	
After completion of course:	
CO1	Recall the basics of internet protocols like HTTP, FTP, and SMTP, and recognize key internet elements such as telnet, usenet, and gopher, as well as security concepts like cookies and firewalls.
CO2	Describe the structure of HTML, including its elements, attributes, and the use of semantic HTML5 tags to create web pages with forms, tables, hyperlinks, and images.

CO3	Utilize cascading style sheets (CSS) to style web pages, including text, backgrounds, and layouts (Flexbox and Grid), and employ frameworks like Bootstrap for responsive web design.
CO4	Implement JavaScript for client-side scripting to handle events, validate forms, manage cookies, and dynamically control web page functionality.

## **BCA-51P-104: Web Application Development Lab**

After completion of course:

CO1	Recall essential web development tools and technologies, including HTML, CSS, JavaScript, and version control systems like Git, along with their usage in creating basic web pages..
CO2	Demonstrate an understanding of HTML5 elements, CSS styling techniques, and JavaScript functions to develop structured and interactive web pages.
CO3	Implement responsive web designs using CSS frameworks like Bootstrap, and use JavaScript to validate forms, handle DOM events, and manage browser storage (cookies and local storage).
CO4	Examine and debug the behavior of web applications using browser developer tools, ensuring cross-browser compatibility and performance optimization.

## **BCA-51T-105: Computer Fundamentals & Office Management Tools**

After completion of course:

CO1	To learn the basic concepts of Computer fundamentals and understand the concept of input and output devices of Computers.
CO2	Differentiate between system and application software and perform basic arithmetic operations using different number systems including binary arithmetic.

CO3	Learn basic word processing, Spreadsheet and Presentation Graphics Software skills.
CO4	Differentiate between system and application software.

## BCA Part I Sem-II

<b>BCA-52T-111: Operating Systems</b>	
After completion of course:	
CO1	Explain fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
CO2	Classify important algorithm e.g. Process scheduling and memory management algorithms.
CO3	Categorize the operating system's resource management techniques, dead lock management techniques, memory management techniques.
CO4	Demonstrate the ability to perform System Administration tasks in LINUX.

<b>BCA-52P-112: Operating Systems Lab</b>	
After completion of course:	
CO1	Explain fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
CO2	Classify important algorithm e.g. Process scheduling and memory management algorithms.
CO3	Categorize the operating system's resource management techniques, dead lock management techniques, memory management techniques.
CO4	Demonstrate the ability to perform System Administration tasks in LINUX.

## **BCA-52T-113: Database Management Systems**

After completion of course:

CO1	Understand the role of database management systems in information technology applications within organizations.
CO2	Effectively explains the basic concepts of databases and data models.
CO3	Develops an Entity-Relationship model based on user requirements.
CO4	Designs SQL queries to create database tables and make structural modifications.

## **BCA-52P-114: DBMS Lab**

After completion of course:

CO1	Understand about SQL Fundamentals.
CO2	Understand about table View, Log &Triggers.
CO3	Learn about Queries using AND- OR- NOT operation, Union- Intersection and Projection, Join Operation, Sorting and Grouping.
CO4	Learn about Nested queries using SQL.

## **BCA-52T-115: Computer Organization & Architecture**

After completion of course:

CO1	Explain the basic of computer architecture, organization and design.
CO2	Explain Data representation, design of sequential and Arithmetic circuits.
CO3	Describe Boolean algebra, data representation and Micro-operations.
CO4	Explain various instructions and addressing modes.

## BCA Part II Sem-III

<b>BCA-63T-201: Data Structures and Algorithms</b>	
After completion of course:	
CO1	Learn the basic types for data structure, implementation and application.
CO2	Know the strength and weakness of different data structures.
CO3	Use the appropriate data structure in context of solution of given problem
CO4	Develop programming skills which require to solve given problem

<b>BCA-63P-202: Data Structures Lab Using C</b>	
After completion of course:	
CO1	Ability to understand a systematic approach to organizing, writing and debugging C programs
CO2	Ability to implement linear and non-linear data structure operations using C programs.
CO3	Ability to solve problems implementing appropriate data structures.
CO4	Ability to implement sorting and searching algorithms using relevant data structures

<b>BCA-63T-203: Object Oriented Programming Through C++</b>	
After completion of course:	
CO1	Explain OOP's features and C++ concepts..
CO2	Construct class and object using constructors.
CO3	Explain and apply the concept of polymorphism and inheritance.
CO4	Execute string functions and handling data files and operations on files.



## **BCA-63P-204: OOP Lab**

After completion of course:

CO1	Describe Object Oriented programming concept.
CO2	Explain the concept of classes, objects, constructors and destructor.
CO3	Demonstrate the concept of inheritance and Polymorphism.
CO4	Describe Object Oriented programming concept.

## **BCA-63T-205: Software Engineering**

After completion of course:

CO1	Understand the fundamentals of software engineering, including key concepts such as software processes, requirement analysis, and various software development models like Waterfall, Spiral, and Agile.
CO2	Identify and explain software project planning techniques, including cost estimation models (like COCOMO), project scheduling, and quality assurance plans (verification and validation).
CO3	Apply design engineering principles to software development, including design concepts such as modularity, abstraction, and architecture, and develop design models for various system elements.
CO4	Analyze software testing strategies and tactics, including unit, integration, system, black-box, and white-box testing, to ensure software reliability and functionality.
CO5	Evaluate software reliability and risk management techniques, and understand the processes involved in software maintenance, reengineering, reverse engineering, and forward engineering.
CO6	Design and implement a comprehensive risk management plan and integrate software reliability measurement techniques to ensure sustainable software development and long-term system reliability.