



**MAHARAJA SURAJMAL BRIJ UNIVERSITY  
BHARATPUR**

**NATIONAL EDUCATION POLICY-2020  
STRUCTURE OF UG-BACHELOR OF COMPUTER  
APPLICATION (BCA)  
Syllabus for III & IV Semester**

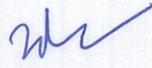
*PS*

*Dr. Arun Kumar Pandey*  
डॉ. अरुण कुमार पाण्डेय  
उपकुलसचिव  
प्रभारी अकादमिक प्रथम

## BCA Part - II (III & IV Semester)

Semester-wise Titles of the Papers in Bachelor of Computer Application						
Diploma in Computer Application						
Semester	Course Code	Course Title	Course Type	Theory/ Practical	Teaching Hours/ Week	Credits
III	BCA 40-T-301	Data Structures and Algorithms	CC(Major)	Theory	4	4
	BCA- 40-P-302	Data Structures Lab Using C	CC(Major)	Practical	4	2
	BCA- 40-T-303	Object Oriented Programming Through C++	CC(Major)	Theory	4	4
	BCA 40-P-304	OOP Lab	CC(Major)	Practical	4	2
	BCA 40-T-305	Software Engineering	CC(Major/ Minor)	Theory	6	6
		MDC-I	MDC	Theory	4	4
		SEC-III	SEC	Theory/ Practical		2
		VAC-III	VAC	Theory/ Practical		2
			<b>Total</b>			
IV	BCA 40-T-401	PHP Programming	CC(Major)	Theory	4	4
	BCA 40-P-402	PHP Lab	CC(Major)	Practical	4	2
	BCA 40-T-403	Object Oriented Concepts Using Java Programming	CC(Major)	Theory	4	4
	BCA 40-P-404	Java Lab	CC(Major)	Practical	4	2
	BCA- 40-T-405	Mathematics & Statistics	CC(Major/ Minor)	Theory	6	6
		MDC-II	MDC	Theory	4	4
		SEC-IV	SEC	Theory/ Practical		2
		VAC-IV	VAC	Theory/ Practical		2
			<b>Total</b>			

*PS*

  
**डॉ. अरुण कुमार पाण्डेय**  
 उपकुलसचिव  
 प्रभारी अकादमिक प्रथम

## Course Content for BCA Semester III and IV

### Semester: III

Course Code: <u>BCA-40-T-301</u>	Course Title: <b>Data Structures and Algorithms</b>
Course Credit : 04	Hours/Week: 04

### Course Objectives (COs):

By the end of this course, students will be able to:

- To design efficient algorithms using various algorithm designing strategies
- To analyze the problem and develop the algorithms related to these problems
- To classify the problem and apply the appropriate design strategy to develop algorithm
- To design algorithm in context of space and time complexity and apply asymptotic notation

### Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand the concepts of data structures and their importance in computing.
- Implement and use common data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
- Analyze the time and space complexity of algorithms related to data structures.
- Apply data structures to efficiently solve real-world problems

## **Data Structures and Algorithms**

### **UNIT – I**

**Introduction to Algorithm Design:** Algorithm, its characteristics, efficiency of algorithms, analyzing Algorithms and problems.

**Linear Structure:** Arrays, records, stack, operation on stack, implementation of stack as an array, queue, types of queues, operations on queue, implementation of queue.

### **UNIT – II**

**Linked Structure :** List representation, Polish notations, operations on linked list - get node and free node operation, implementing the list operation, inserting into an ordered linked list, deleting, circular linked list.

PS

**Tree Structure:** Concept and terminology, Types of trees, Binary search tree, inserting, deleting and searching into binary search tree, tree traversals.

### UNIT – III

**Graph Structure:** Graph representation - Adjacency matrix, adjacency list, Warshall's algorithm, adjacency multilist representation. Orthogonal representation of graph . Graph traversals - BFS and DFS. Shortest path, transitive closure.

### UNIT – IV

**Searching and sorting:** Searching - sequential searching, binary searching, hashing. **Sorting** - selection sort, bubble sort, quick sort, heap sort, merge sort, and insertion sort, efficiency considerations.

#### Recommended reference books

1. S.Lioschutz: Data Structures, Mc Graw Hill International Edition.
2. A.V.Aho., J.E.Hopcroft, and J.D.Ullman, Data Structures and Algorithms, Pearson.
3. A. MichaelBerman: Data Structures via C++, OxfordUniversity Press.
4. SaraBaase and AllenVan Gelder: Computer Algorithms, Pearson Education Asia.

Course Code: <b>BCA-40P-302</b>	Course Title: <b>Data Structures Lab Using C</b>
Course Credit : <b>02</b>	Hours/Week: <b>04</b>

#### Content: Suggested Experiments

1. To write a C program for Implementation of stack using array.
2. To write a C program for Implementation of queue using array.
3. Write a C program that uses functions to perform the following on Singly Linked List:  
i) Creation ii) Insertion iii) Deletion iv) Traversal
4. To Write a C program that Implementation of Binary Trees and operations of Binary Trees.
5. To write a C program for implementation of Binary Search tree.
6. To Write a C program that Implementation of searching techniques.
7. To Write a C program to implement bubble sort.
8. To Write a C program to implement merge sort.

*PS*

Course Code: <b>BCA-40-T-303</b>	Course Title: <b>Object Oriented Programming Through C++</b>
Course Credit : 04	Hours/Week: 04

### Course Objectives (COs):

Upon finishing the course, students will be able to:

- Introduction to Object-Oriented Programming: Understand the fundamental concepts of object-oriented programming (OOP) and how it differs from procedural programming.
- Java Syntax and Semantics: Learn the basic syntax and semantics of the Java programming language, including data types, operators, and control structures.
- Classes and Objects: Understand the principles of classes and objects in Java. Learn how to define and instantiate classes, and how to use objects to encapsulate data and methods.
- Inheritance and Polymorphism: Explore the concepts of inheritance and polymorphism in Java. Understand how to create class hierarchies and use polymorphic behavior to enhance code flexibility and reusability.
- Encapsulation and Abstraction: Learn about encapsulation and abstraction principles. Understand how to use access modifiers to protect data and methods, and how to define abstract classes and interfaces.
- Exception Handling: Understand the mechanisms for error and exception handling in Java. Learn how to use try-catch blocks, create custom exceptions, and manage resources with the try-with-resources statement.

**Course Outcomes (COs):** On completion of the course, the student will be able to:

- Understand the fundamental concepts of object-oriented programming.
- Design and implement C++ programs using classes, objects, and inheritance.
- Apply polymorphism and templates to develop reusable code.
- Utilize advanced features of C++ to develop efficient and modular programs.

*PS*

# Object Oriented Programming Through C++

## UNIT – I

**Introduction to Object Oriented Concepts:** Evolution of OOP, OOP Paradigm, advantages of OOP, comparison between functional programming and OOP approach, characteristics of object oriented language – objects, classes, inheritance, reusability, user defined data types, polymorphism, overloading.

## UNIT – II

**Introduction to C++:** C++ tokens, data types, C++ operators, type conversion, variable declaration, arrays, statements, expressions, conditional statements, Jumping statements, loops, functions, pointers, structures.

**Classes and Objects:** Classes, objects, defining member functions, arrays of class objects, pointers and classes, passing objects, constructors, types of constructors, destructors, this pointer, access specifiers, friend functions, inline functions.

## Unit – III

**Inheritance:** Introduction, Importance of Inheritance, types of inheritance, Constructor and Destructor in derived classes., member access control.

**Polymorphism:** Functions Overloading, Operator Overloading, early binding polymorphism with pointers, Unary and Binary Operator Overloading, Overload Assignment Operator, Copy Constructor.

## Unit –IV

**Virtual Function :** Virtual Function, late binding, pure virtual functions, abstract classes, Generic Programming with Templates, Friend function, Overloaded Function Templates, Multiple Arguments function Template.

**File Management:** Handling Data files (sequential and random), Opening and closing of files, stream state member functions, Operations on File, Exception Handling.

### Recommended Books:

1. Deitel HM & Deitel JP; C/C++ How to program; 5<sup>th</sup>Edn; Pearson Pub.
2. Balagurusamy ; Object Oriented Programming in C++; 4<sup>th</sup> Edition TMH.
3. Venugopal, Rajkumar; Mastering C++; Tata Mcgrow Hill.
4. KanetkarY.: LET US C++; BPB;
5. ByronGottfried; Programming with C;TMH;



Course Code: <b>BCA-40P-304</b>	Course Title: <b>OOP Lab</b>
Course Credit : 02	Hours/Week: 04

**Contents: Suggested Experiments**

1. Write a c++ program to check whether given number is Palindrome or not.
2. Write a c++ program to implement Functions
3. Write a c++ program to implement Bank account Class
4. Write a c++ program to implement Student information Class
5. Write a c++ program to implement Constructors.
6. Write a c++ program to find Factorial of a given number using Recursion
7. Write a c++ program to implement Friend function
8. Write a c++ program to implement Function Templates
9. Write a c++ program to implement Multiple inheritance
10. Write a c++ program to implement Hierarchical inheritance
11. Write a c++ program to implement Function overloading
12. Write a c++ program to implement Exceptional handling
13. Write a c++ program to implement Class Templates.
14. Write a c++ program to implement Virtual Functions
15. Write a program to copy the contents of one file to another.
16. Write a program to perform read/write binary I/O operation on a file (i.e. write the object of a structure/class to file).

Course Code: <b>BCA-40T-305</b>	Course Title: <b>Software Engineering</b>
Course Credit : 06	Hours/Week: 06

**Course Objectives (COs):**

By the end of this course, students will be able to:

- Understand the principles and practices of software engineering.
- Apply software engineering processes and methodologies to develop software systems.
- Perform requirements analysis and software design.
- Implement software using appropriate programming languages and development tools.
- Apply software testing and quality assurance techniques.

PS

## Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand the principles and practices of software engineering.
- Apply software engineering processes and methodologies to develop software systems.
- Perform requirements analysis and software design.
- Implement software using appropriate programming languages and development tools.
- Apply software testing and quality assurance techniques.

## Software Engineering

### Unit-I

**Software Engineering Fundamentals:** Software, Problem Domain, Software Engineering Challenges, Software Processes (processes, projects & products, component), Software Requirement Analysis & Specification.

**Software Development Process Models:** Waterfall Model, Prototyping, Iterative Enhancement Model, Spiral Model. Introduction to Agile Model: Principles, Steps, Various Agile Process Models.

### Unit-II

**Software Project Planning:** Cost Estimation- Uncertainties in Cost Estimation, Building Cost Estimation Models, On Size Estimation, COCOMO Model.

**Project Scheduling:** Average Duration Estimation, Project Scheduling & Milestones. Quality Assurance Plans: Verification & Validation, Inspection & Reviews.

### Unit-III

**Design Engineering:** Design Process & Design Quality, Design Concepts (abstraction, architecture, modularity, functional independence, refinement, and design classes), The Design Model (data design elements, architectural design elements, interface design elements, component-level design elements, deployment-level design elements).

**Testing Strategies & Tactics:** A strategic approach to software testing, Strategic issues, Software testing fundamentals, Test characteristics, Test Strategies for conventional software: Unit Testing, Integration testing, Validation Testing, System testing, Black-Box testing, White Box testing.

### Unit-IV

**Software Reliability:** Risk Management, Measures of Reliability & Availability, Software Safety.

**Maintenance and Reengineering:** Introduction to: Software Maintenance, Software Supportability, Reengineering, Reverse Engineering, Restructuring, and Forward Engineering.



### Reference /Text Books

1. Pressman, Roger (2001) Software Engineering; A Practitioner's Approach, 8<sup>th</sup> ed. M Graw-Hill,2014.
2. Sommerville Ian; Software Engineering, 9<sup>th</sup>Ed.Pearson Education,2014
3. Jalote, Pankaj (7) An integrated Approach to Software Engineering 2<sup>nd</sup>Ed.
4. JamesRumbaugh. MichealBlaha, "Object oriented Modeling and Design with UML", 2<sup>nd</sup> Edition, 2007.
5. SimonBennett, SteveMcRobb and RayFarmer, " Object-Oriented Systems Analysis and Design Using UML" 4<sup>th</sup>Edition,McGraw Hill Education, 2010

PS

## Semester: IV

Course Code: BCA - 40-T-401	Course Title: PHP Programming
Course Credit : 04	Hours/Week: 04

### Course Objectives (COs):

In this course, you will learn about:

- Fundamental concepts of PHP scripting language
- The basic structure of a web application
- The request/response cycle
- Basics of MySQL database
- The relationship between the client-side and server-side scripts
- Creating functional websites and web apps in PHP
- PHP web application testing and security
- Creating a PHP web application using a CMS

### Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Able to learn Core-PHP, Server Side Scripting Language.
- Able to design dynamic and interactive web pages, websites.
- Able to run PHP scripts on server and retrieve results.
- Able to handle databases like MySQL using PHP in web sites.

## PHP Programming

### UNIT-I

**Introduction to PHP:** Installation of PHP and MySQL, PHP configuration in IIS & Apache Web Server. Features of PHP, Writing PHP, Parsing PHP code, Embedding PHP and HTML Executing PHP and viewing in Browser.

### Unit - II

**Control Structures:** Data types, Operators, PHP variables: static and global variables, Comments in PHP, Control Structures, Condition statements, If...Else, Switch, ? operator, Loops, While, Break Statement Continue. Do...While, For, For each, Exit, Die, Return. Arrays: Numeric, Associative and Multidimensional Arrays

*PS*

### UNIT-III

**Strings:** Creating and accessing String, Searching & Replacing String, Formatting String, String Related Library function, Pattern matching, Replacing text, Splitting a string with a Regular Expression

**Functions:** Defining a Function, Calling a Function, Parameter passing, Returning value from function

**Form Data Handling:** \$\_GET, \$\_POST, \$\_REQUEST Variables, Cookies handling, Session Management

### UNIT-IV

**Exception Handling:** Understanding Exception and error, Try, catch, throw

**File Handling:** Opening and closing a file, Copying, renaming and deleting a file

**Database Handling:** Connection with MySql Database or ODBC, Performing basic database operation (Insert, Delete, Update, Select, Truncate Alias, Order By), Setting query parameter.

#### References

1. PHP, The Complete Reference, Steven Holzner, TMH
2. Beginning PHP 5.3, Matt Doyle, John Wiley & Sons
3. Core PHP Programming Leon Atkinson Pearson publishers
4. Beginning PHP 5.0 Database Christopher Scollo, Harish Rawat, Deepak Thomas, Wrox Press

Course Code: <u>BEA-40P-402</u>	Course Title: <b>PHP Lab</b>
Course Credit : <b>02</b>	Hours/Week: <b>04</b>

#### Content: Recommended Experiments:

1. The sum of digits of given number
2. Check whether the given number is prime or not
3. The factorial of a given number
4. Fibonacci series of a given number
5. Get name of the user from a form and show greeting text.
6. To check whether given number is palindrome or not.
7. To find largest values of two numbers using nesting of function.
8. Write a Mathematical calculator program.
9. To check whether given number is String palindrome or not.
10. Create a PHP page for login page without sql connection.
11. Write a php program to Array manipulation.
12. Create a PHP page for login page with sql connection.
13. Write a php program to Read from existing file.
14. Write a php program to write a file.

PL

Course Code: <b>BCA-407-403</b>	Course Title: <b>Object Oriented Concepts Using Java Programming</b>
Course Credit : 04	Hours/Week: 04

### **Course Objectives (COs):**

By the end of this course, students will be able to:

- Learn the Object-Oriented Programming concepts to write, compile and debug programs using Java language.
- Apply the concepts of object-oriented programming like polymorphism, inheritance, Exception Handling, and Multithreading.
- Design and develop console and GUI applications using Java Programming Language.
- Work on programming project as individual or as team member in design, development and implementation phase.

### **Course Outcomes (COs):**

After completing this course satisfactorily, a student will be able to:

- Understand the features of Java and the architecture of JVM
- Write, compile, and execute Java programs that may include basic data types and control flow constructs and how type casting is done
- Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism and inheritance
- The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA's Exceptional handling mechanism compared to other Programming Language
- Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

*PS*

# Object Oriented Concepts Using Java Programming

## UNIT – I

**Java Programming :** Basic concepts of object oriented programming(Objects and Classes, Data Abstraction & Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message passing), Java features, JVM, Byte code interpretation, simple java program, command line argument, Data types, type casting, operators (Arithmetic, increment, decrement, relational, logical, bit wise, conditional) and expressions.

## UNIT – II

**Decision Making and Branching :** Decision making and branching (if...else, else if, switch), looping, classes, objects and methods, visibility control, constructors, wrapper classes, nesting of methods, Arrays and strings handling. **Polymorphism:** Function overriding, Operator overloading, final classes.

## UNIT – III

**Inheritance & Multithreaded Programming :** Inheritance, Types of Inheritance, Abstract class, interfaces, packages, multithreaded programming, extending thread, life cycle of thread, using thread methods, thread priority, synchronization.

**Exception Handling :** Exception-Handling fundamentals, Exception types, try, catch, throw, finally, creating exception sub classes.

## UNIT – IV

**JSP:** Introduction to JSP, Directory Structure, Lifecycle JSP, Scripting Elements. JAR files, Servlets Life cycle of servlet, JDBC connectivity.

### Recommended Text Books

1. Mastering java 2 ", BPB Publications. Programming with Java A Primer, E.Balagurusamy Tata McGraw Hill Companies
2. Java Programming JohnP.FlyntThomson2nd
3. The complete reference JAVA2, Herbertschildt. TMH
4. Arnold,Gosling, " The Java Programming Professional 2000", AddisonWesley Publication
5. C.Thomaswu, "An introduction to oop with Java", TMH

Course Code: <b>BCA - 40 P - 404</b>	Course Title: <b>Java Lab</b>
Course Credit : <b>02</b>	Hours/Week: <b>04</b>

### Content: Suggested Experiments

1. Installation of Java software, study of any integrated development environment
2. Create a test project, add a test class and run it.



3. Practice Java Basic Programs on Classes and Objects.
4. Write a java program to computer water consumption bill.
5. Write a java program to illustrate the concept of class with method overloading.
6. Write a java program to illustrate the concept of class with Constructors overloading.
7. Write a program to create a class named shape. It should contain 2 methods, draw() and erase() that prints "Drawing Shape" and "Erasing Shape" respectively. For this class, create three sub classes, Circle, Triangle and Square and each class should override the parent class functions - draw () and erase (). The draw() method should print "Drawing Circle", "Drawing Triangle" and "Drawing Square" respectively. The erase() method should print "Erasing Circle", "Erasing Triangle" and "Erasing Square" respectively. Create objects of Circle, Triangle and Square in the following way and observe the polymorphic nature of the class by calling draw() and erase() method using each object. Shape c=new Circle(); Shape t=new Triangle(); Shape s=new Square();
8. Write a Java Program to implement multilevel inheritance.
9. Write a Java program to implement the method overriding.
10. Write a Java program to implement abstract class.
11. Write a Java Program to implement Packages.
12. Write a Java Program to implement Access Protection in Packages.
13. Write a Java program to demonstrate interfaces.
14. Write a Java program to implement the exception handling mechanism.
15. Write a Java Program to demonstrate the following String Handlings.
  - i. String Length& Concatenation. ii. Character Extraction. iii. String Comparison.
16. Write a Java program for multi-thread implementation.

Course Code: <b>BCA-40-T-405</b>	Course Title: <b>Mathematics &amp; Statistics</b>
Course Credit : 06	Hours/Week: 06

### Course Objectives (COs):

By the end of this course, students will be able to:

- Be able to apply problem-solving and logical skills
- Have a deeper understanding of mathematical theory
- Have a solid knowledge of elementary statistics
- Be able to communicate mathematical/logical ideas in writing
- Be competent in computer programming
- Be familiar with several subfields of mathematics (e.g, numerical analysis, topology, operations research).
- Be exposed to undergraduate research or internship opportunities

*PS*

## Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- To understand the basic concepts of Mathematical reasoning, set and Relations and functions.
- To understand the basic concepts of functions and logic/proofs.
- To understand various counting techniques.
- Understand the concepts of various types of Matrices and Determinants.
- To understand the concept of Statistics: Data collection and Classification.
- To understand the concept of measurement of central tendency.
- To understand the correlation and regression analysis.

## Mathematics & Statistics

### UNIT – I

**Sets :** Definition of sets, representation of sets, type of sets, Operations on sets, Sub sets, Power set, Universal set, Complement of a set, Union and Intersection of two sets, Venn diagrams, Principles of Inclusion and Exclusion. **Relations:** Cartesian product of sets, Definition of relation, Types of relations- reflexive, symmetric, anti-symmetric, transitive, equivalence.

**Functions :** Definition, Domain & Range of a functions, one to one and onto functions, Bijective functions, composite functions, inverse of functions.

### UNIT – II

**Logic and Proofs :** Proposition, Conjunction, Disjunction, Negation, Compound proposition, De Morgan's laws, Tautology and Contradiction.

**Matrices:** Definition and Types of Matrices, Addition, Subtraction and Multiplication of Matrices, Non-commutativity of multiplication of matrices, Scalar Multiplication, Transpose of a Matrix.

**Determinant:** Determinant of a square matrix (up to  $3 \times 3$  matrices), properties of determinants, minors, cofactors, expansion of determinants, application of determinants in finding the area of a triangle. Adjoint and Inverse of a matrix, Solution of system of linear equations by Cramer's Rule.

### UNIT –III

**Statistics:** Data collection methods, Data classification, Frequency Distribution, Graphical representation of frequency distribution. **Measures of Central Tendency-** Mean, Median, Mode, **Measures of Dispersion-** Mean Deviations, Standard Deviations, Variance

### .UNIT –IV

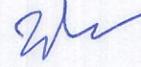
**Correlation Analysis :** Correlation, Types of Correlations, Methods of Studying Correlations, Measure of Karl Pearson's coefficient of correlation, Rank Correlation Coefficient.

**Regression Analysis:** Regression, Use of regression analysis,, Difference between Correlation and Regression Analysis, Regression Lines Equations, Properties of regression lines.



### Reference Books:

1. C.L.Liu: Elements of Discrete Mathematics, Tata Mc-Graw Hill Publishing Company Ltd., 2000
2. Seymour Lipschutz; Discrete Mathematics;TMH.
3. KennethHRosen; Discrete Mathemtics& Its Applications; 6 Edition,MGH;
4. RichardJohnsonbaugh: Discrete Mathematics, Pearson Education, Asia, 2001
5. JohnTruss: Discrete Mathematics for Computer Scientists, Pearson Education, Asia, 2001.
6. Basic Mathematics, R.D.Sharma
7. B.L.Agrawal; Basic Statistics; KhannaPub.
8. S.P.Gupta; Statistical Methods; Sultan Chand & Sons
9. S.C.Gupta, V.K.Kapoor ; fundamental of statics; Sultan Chand & Sons



•• अरुण कुमार पाण्डेय  
उपकुलसचिव  
प्रभारी अकादमिक प्रथम