# Shri Vaishnav Vidhyapeeth Vishvavidhyalaya, Indore

# Institute of Computer Applications

## Name of Program : BCA + MCA

| COURSE CODE | CATEGORY | COURSE NAME | L | T | P | CREDITS | TEACHING & EVALUATION SCHEME | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | THEORY | | | PRACTICAL | |
| | | | | | | | END SEM University Exam | Two Term Exam | Teachers Assessment* | END SEM University Exam | Teachers Assessment* |
| BCCA202 | COMPULSORY | Mathematical Foundation of Computer Science - II | 3 | 1 | 0 | 4 | 60 | 20 | 20 | 0 | 0 |

**Legends:** L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;
Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.
**\*Teacher Assessment** shall be based on following components: Quiz/Assignment/Project/Participation in class (Given that no component shall be exceed 10 Marks)

### Course Educational Objectives(CEOs):
- To introduce the students with the concepts of Probability and Distributions.

### Course Outcomes (Cos:)
- After the successful completion of this course students will be able to understand and apply the fundamentals of probability.
- Students will be able to demonstrate the use of distributions.
- Students will be able to apply the hypothesis in test cases.

### UNIT – I
Probability: Axioms, Conditional probability, Baye's theorem, Random Variables, Discrete random variables – Binomial, Geometric, Poisson RV.

### UNIT – II
Continuous RV – Uniform, Exponential, Gamma & Normal,

### UNIT – III
Expectation, Mean and Variance, Jointly distributed RVs, Co-variance, Sums of RVs, Central Limit theorem, Moment Generating functions.

### UNIT – IV
Sample Distribution, Inference concerning mean, Statistical Inference – Parameter estimation, Maximum likely-hood estimation

**UNIT – V**
Testing of Hypothesis, Curve Fitting-Method of the Least Square, Curvilinear multiple regression

**Suggested Readings:**
1. Ross, S: A First course in probability, sixth edition, Pearson Education.
2. Ross Sheldon: Introduction to Probability models, Eighth edition, Elsevier, 2003
3. Trivedi K. S.: Probability & Statistics with Reliability, Queuing and Computer Science Applications, Second edition, Wiley, 2002

# Name of Program : BCA + MCA

| COURSE CODE | CATEGORY | COURSE NAME | L | T | P | CREDITS | END SEM University Exam | Two Term Exam | Teachers Assessment* | END SEM University Exam | Teachers Assessment* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | THEORY | | | PRACTICAL | |
| BCCA204 | COMPULSORY | Data Structure using C | 3 | 1 | 0 | 4 | 60 | 20 | 20 | 0 | 0 |

**Legends:** L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;
Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.
**\*Teacher Assessment** shall be based on following components: Quiz/Assignment/project/Participation in class (Given that no component shall be exceed 10 Marks)

**Course Educational Objectives(CEOs):**
- To understand the students with the applications of Standard data structure in real world problems.
- To provide knowledge of creation of new data structures.
- To familiarize the students with the analysis and design a particular problem.

**Course Outcomes (Cos):students will be able to**

- Demonstrate familiarity with major algorithms and data structures.
- Analyze performance of algorithms.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, trees and graphs
- Demonstrate understanding of various sorting algorithms, including bubble sort, insertion sort, selection sort, heap sort and quick sort.
- Understand and apply fundamental algorithmic problems including Tree traversals, Graph traversals, and shortest paths.
- Demonstrate understanding of various searching algorithms.

- Program multiple file programs in a manner that allows for reusability of code.
- Compare different implementations of data structures and to recognize the advantages and disadvantages of the different implementations.
- Write complex applications using structured programming methods.

## UNIT - I
The Concept of Data Structure, Abstract Data Type, Concept of List and Array, Introduction to Stack, Stack as an Abstract Data Type, Primitive Operations on Stack, Stack's Applications -  polish notations Infix, Postfix, Prefix and Recursion. Introduction to Queues, Primitive operations on Queues, Queue as an Abstract Data Type, Circular Queue, Dequeue, Priority Queue, Applications of Queue.

## UNIT - II
**Linked List -** Introduction to Linked List, Memory Representations of Linked List, comparison; Operations on Linked List, Linked Representation of Stack and Queue, Header Nodes. Types of Linked List : Doubly Linked List, Circular Linked List, Application of Linked List, Self Referential Structures.

## UNIT –III
**Trees:** Definition, Basic Terminology of Trees, Tree Representations as Array and Linked. Binary Trees, Binary Tree Operations. Traversal of Binary Trees – In order, Preorder & Post order, Complete Binary Tree, almost complete binary tree; Application of Binary Tree.

## UNIT-IV
**Complexity:** concept and notations. Searching: Sequential, Binary and their comparison. Sorting - External and Internal Sorting, Insertion Sort, Selection Sort, Quick Sort, Bubble Sort, Heap Sort, Comparison of Sorting Methods. Hashing;

## UNIT-V
**Graphs** - Introduction to Graphs, Basic Terminology, Directed, Undirected and Weighted graph, Representation of Graphs, Graph Traversals - Depth First and Breadth First Search. Applications of Graphs.

**Suggested Readings:**
1. Kruse R.L. Data Structures and Program Design in C; PHI
2. Aho, "Data Structure & Algorithms".
3. Trembly  and *SORRENSON,*"Introduction to Data Structure with Applications".
4. TennenBaum A.M. & others,"Data Structures using C & C++"; PHI
5. Horowitz & Sawhaney,"Fundamentals of Data Structures", Galgotia Publishers.
6. Yashwant Kanetkar, "Understanding Pointers in C", BPB.
7. Lpschuists, "Data Structure", (Schaum 's Outline Series, McGraw Hill publication)
8. Ellis Horowitz and Sartaj Sawhney, "Fundamentals of Computer Algorithm"

Chairperson
Board of Studies
(Computer Science & Engineering,
Information Technology & Computer Applications)
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

Joint Registrar
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore

# Shri Vaishnav Vidhyapeeth Vishvavidhyalaya, Indore

# Institute of Computer Applications

## Name of Program: BCA + MCA

| COURSE CODE | CATEGORY | COURSE NAME | L | T | P | CREDITS | TEACHING & EVALUATION SCHEME | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | THEORY | | | PRACTICAL | |
| | | | | | | | END SEM University Exam | Two Term Exam | Teachers Assessment* | END SEM University Exam | Teachers Assessment* |
| BCCA205 | COMPULSORY | Object Oriented Methodology & C++ | 3 | 1 | 0 | 4 | 60 | 20 | 20 | 0 | 0 |

**Legends:** L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical; C - Credit; Q/A – Quiz/Assignment/Attendance, MST - Mid Semester Test.

**\*Teacher Assessment** shall be based on following components: Quiz/Assignment/Project/Participation in class (Given that no component shall exceed 10 Marks).

**Course Educational Objectives (CEOs):**

- To familiarize the students with programming and to encourage them to develop their logic.
- To make students well versed with C++ language to solve problems efficiently.
- Using simple and well drawn illustrations develop their programming skills using modular programming.

**Course Outcomes (COs):** Student will be able to:

- Develop algorithms for problems.
- To understand the object oriented paradigm.
- Apply the programming concepts to solve the given problems.
- Write the programs using modular programming.
- Understand and write programs using various data structures very efficiently.
- Write the programs using pointers and to manage memory.
- To apply the knowledge of Object Oriented Methodology to write reusable code.
- Implement programs of file handling.

### UNIT-I
**Object Oriented Paradigm:** Need, Features, comparison with other programming paradigm, various programming styles; Basic concepts: C++ basics, C++ character set, C++ Tokens (Identifiers, Keywords, Constants), operators, precedence and associativity, expressions, type conversion and type casting, Structure of a C++ Program (include files, main function), cout, cin, loops and decisions.
Use of I/O Operators (<< and >>), Cascading of I/O Operators, Inline Functions, Reference Variables, Call by Reference, Function Overloading.

**Unit II:**
Concept of a class, Data Members and Member Functions, visibility modes, scope resolution operator (::); Objects: accessing members through object, object and classes, object arrays, passing objects as function arguments, Friend Function; Constructor and Destructor functions: Default Constructor, Overloaded Constructors, Copy Constructor, Constructor with default arguments, Dynamic Constructors.

**UNIT-III**
Operator overloading; Inheritance: Concepts of Base Class and Derived Class; Private, Public and Protected derivation of classes, accessibility of base class members by objects of derived class, access mechanism in classes, ambiguity resolution in inheritance, virtual base classes, constructors in derived classes, member/nested classes.

**UNIT-IV**
**Pointers:** Declaration and Initialization of Pointers; Dynamic memory allocation, new, delete operators; Pointers and Arrays: Array of Pointers, Pointer to an Array, Pointer to Objects, Array of Pointers to Objects, this Pointer, Pointers to Derived Classes, Virtual Functions, Early and Late Binding , Runtime Polymorphism, Pure Virtual Functions, Abstract Classes.

**UNIT-V**
C++ I/O system, Use of defined manipulators, formatted I/O, creating inserters and extractors, file I/O basics, Data File Handling:  creating disk files and file manipulations using seekg(), seekp(), tellg() and tellp() functions, if stream, of stream, file stream classes; Exception Handling: Try, Catch and Throw

**Suggested Readings:**
1. Robert Lafore, "Object-Oriented Programming in C++", 4e, SAMS.
2. Ricahrd C. Lee, Wiiliam M. Tepfenhart "UML & C++: A Practical Guide To Object Oriented Development", 2e, Pearson.
3. Herbert Schildt, "C++ the complete reference", 4e, 2003.
4. E Balagurusamy, "Object Oriented Programming with C++", 6e, McGraw Hill Education.
5. G.Booch, "Object Oriented Analysis And Design With Applications", 3e, Pearson.
6. K.R. Venugopal, Rajkumar Buyya, "Mastering C++", 2e, McGraw Hill Education.

## Name of Program : BCA + MCA

| COURSE CODE | CATEGORY | COURSE NAME | L | T | P | CREDITS | THEORY END SEM University Exam | THEORY Two Term Exam | THEORY Teachers Assessment* | PRACTICAL END SEM University Exam | PRACTICAL Teachers Assessment* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BCCA206 | COMPULSORY | Lab- 1 (Programming Lab in C++) | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 30 | 20 |

**Legends:** L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;
Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.
**\*Teacher Assessment** shall be based on following components: Quiz/Assignment/Project/Participation in class (Given that no component shall be exceed 10 Marks)

**Course  Educational Objectives (CEOs):**
- To familiarize the students with programming and to encourage them to develop their logic.
- To make students well versed with C++ language to solve problems efficiently.
- Using simple and well-drawn illustrations develop their programming skills using modular programming.

**Course Outcomes (COs):** Student will be able to:
- Develop algorithms for problems.
- Apply the programming concepts to solve the given problems.
- Write the programs using modular programming.
- Understand and write programs using various data structures very efficiently.
- Write the programs using pointers and to manage memory.
- Implement programs of file handling.
- Compile and debug the programs
- Understand different error messages and to rectify them

**List of Programs:**
1. Write down the features of Object Oriented Programming?
2. Write down the advantages and disadvantages of OOP's?
3. Define copy constructor. Explain its significance. Under which condition is it invoked? Support your answer with an example.
4. Explain the purpose of function overloading. Write a C++ program that uses afunction to check whether a given number is divisible by another number or not. Give proper comments whether the denominator number is a prime number or not.
5. Define a friend function. Explain the relation of friend function with respect to Public, Private and        Protected data members of the class.
6. Write a program in C++
a. To generate the following number pyramid
1.   0

current date.

24. Create a class contain Book no, Book Title, Book price. The class also store and display the complete information of book. **Total_cost()** function calculate the cost of for **N** number of copies, where **N** is passed to the function as argument and **purchase()** function to ask the user it input the number of copies to be purchased. It invoke **Total_cost()** and print the total cost to be paid by the user.
25. Define a class string use overloaded = = operator to compare two string.
26. Find the largest value of two objects using > operator.
27. Write a C++ program to print the number between 1 to 100 and 100 to 1 using operator overloading.
28. Define a class String that could work as a user defined string type. Include a constructors that will enable us to create an uninitialized string String s1; and also to initialize an object with a string constant at the time of creation like String s2("Welcome"); Include a function that adds two strings to make a third string. Note that the statement s2=s1 ;will be perfectly reasonable expression to copy one string to another .
29. Write a complete program to test your class to see that it does the following tasks:
    (a) Create uninitialized string objects.   (b) Creates objects with string constants
    (c)  Concatenates two string                      (d) Display a desired string objects

## Name of Program : BCA + MCA

| COURSE CODE | CATEGORY | COURSE NAME | L | T | P | CREDITS | TEACHING & EVALUATION SCHEME | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | THEORY | | | PRACTICAL | |
| | | | | | | | END SEM University Exam | Two Term Exam | Teachers Assessment* | END SEM University Exam | Teachers Assessment* |
| BCCA207 | COMPULSORY | Lab- II (Data Structure Lab in C/C++) | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 30 | 20 |

**Legends:** L - Lecture; T - Tutorial/Teacher Guided Student Activity; P – Practical;   C - Credit;
Q/A – Quiz/Assignment/Attendance, MST - Mid Sem Test.
**\*Teacher Assessment** shall be based on following components: Quiz/Assignment/Project/Participation in class (Given that no component shall be exceed 10 Marks)

**Course Educational Objectives (CEOs):**
- To familiarize the students with programming and to encourage them to develop their logic.
- To make students well versed with C++ language to solve problems efficiently.
- Using simple and well drawn illustrations develop their programming skills using modular programming.

**Course Outcomes (Cos):**Student will be able to:
- Develop algorithms for problems.

- Apply the programming concepts to solve the given problems.
- Write the programs using modular programming.
- Understand and write programs using various data structures very efficiently.
- To choose a suitable data structure for a given problem.
- Write the programs using pointers and to manage memory.
- Implement programs of file handling.

**List of Programs:**

1. Write a program to create a two dimensional array and perform add, subtract and multiplication operations.
2. Write a program to create a two dimensional array using dynamic memory allocation.
3. Write a program to implement stack.
4. Write a program to convert infix expression into postfix expression.
5. Write a program to check balanced parentheses for a given infix expression.
6. Write a program to evaluate postfix expression.
7. Write a program to implement queue.
8. Write a program to implement circular queue.
9. Write a program to implement link list with insert, delete, search, view, and
10. delete function.
11. Write a program to implement ordered link list.
12. Write a program to add two polynomials.
13. Write a program to create doubly link list.
14. Write a program to implement tree with insert, delete and search function.
15. Write a program for in order, post order and preorder traversal of tree.
16. Write a program for binary search and sequential search using recursion.
17. Write a program for bubble sort and sequential search.
18. Write a program for insertion sort and quick sort.