St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – II of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2024-25.

DEPARTMENT OF COMPUTER SCIENCE

BCA (Hons.)
Category – IV

Major Course – 1: BCAMC221C Object Oriented Programming with C++

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Object Oriented Programming with C++ (Theory) BCAMC221C	4	0	0	10 + 2 from a recognized board in any stream	Nil

Learning Outcomes:

At the end of the syllabus students will be able:

- To differentiate the procedural programming and object-oriented programming.
- To describe the fundamentals of object-oriented programming such as class, objects, polymorphism, inheritance, encapsulation etc.
- To describe the working with variables and to discuss various types of functions and data members.
- To describe how to initialize variables inside class using constructors and also the usage of destructors.
- To demonstrate the allocation and deallocation of memory in C++.
- To understand the use of advanced features of C++ such as templates, Virtual function, function overloading and operator overloading.

UNIT 1 **Overview of Object-Oriented Programming** **15 Hours**

- Introduction to Object Oriented Programming
- Procedure Oriented and Object Oriented

- Difference Between C and C++
- C++ Output/ Input
- Keywords in C++
- New style of header file specification
- Comments in C++
- Variables in C++
- Reference Variables in C++
- The bool Data type
- Importance of function prototyping in C++
- Function Overloading
- Default Arguments
- Inline Function
- Scope Resolution Operator

Classes And Object

- Structures in C
- Structure in C++
- Access Specifier
- Classes
- Objects in C++
- Characteristics of Access Specifier
- Function outside a class
- Initialization of variable in C++
- Arrow Operator
- 'this' pointer

UNIT 2 **More on Classes and Objects** **15 Hours**

- Member Functions and Data Members
- Friend Functions
- Friend Class
- Array of Class Object
- Passing Class Objects to Function
- Returning Objects from Functions
- Nested Classes
- Namespaces

Dynamic Memory Management

- Introduction
- Dynamic Memory Allocation Using "new"
- Dynamic Memory Deallocation
- "Set_New_Handler" Function

Constructor and Destructor

- Constructor
- Characteristics of Constructor
- Types of Constructors
- Destructor
- Characteristics of Destructor

UNIT 3 **Inheritance** **15 Hours**

- Introduction
- Advantages of Inheritance
- 'Protected' Access specifier
- Inheritance using different access specifier
- Initialization of Base class members through derived class object
- Different forms of Inheritance
- Function Overriding

Virtual Functions and Inheritance

- Introduction
- Pointers to derived class
- Rules for virtual function
- Internals of Virtual Functions
- Pure virtual function
- Virtual Base class
- Virtual destructor
- Abstract class
- Limitations of virtual Function

Early binding v /s Late binding

UNIT 4 **Operator Overloading** **15 Hours**

- Introduction
- Operators that can be overloaded
- Overloading Unary Operator using member Functions
- Overloading Unary Operator using friend Functions
- Overloading Binary Operator using member Functions
- Overloading Binary Operator using friend Functions
- Why Overload Operators using friend function?
- Rules for Operator Overloading

Constructor- Destructor Invocation

- Introduction
- Order of Invocation of Constructors and destructors
- Destructors in Action
- Type Conversions

Templates

- Introduction
- Function Templates
- Function Templates with multiple parameters
- Overloading Function Template
- Class Template
- Class Template with multiple parameters
- Nested Class Templates
- Advantages of using Templates

Textbook:

1. Object Oriented Programming with C++, Publication: Pearson By Subhash KU

2. Reference Books:

1. Object-Oriented Programming with C++ (Second Edition)
   Publication: PHI
   By Poornachandra Sarang

2. Object Oriented Programming using C++
   Publication: Cengage Learning
   By Joyce Farrell

3. Object Oriented Programming In C++
   Publication: Wiley India edition
   By Rajesh K. Shukla

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – II of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

BCA (Hons.) Category – IV

Major Course – 1: BCA-2501 Object-Oriented Programming with C++ (Practicals)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Object Oriented Programming with C++ (Practical) BCAMC222L		0	4	BCA Semester 1	Programming Basics

Learning Objective:

- The objective of this subject is to get in-depth practical knowledge of C++ language.
- To obtain practical knowledge of programming for real life applications.

Learning Outcome:

At the end of the course, the student will be able to:

- Develop solutions for a range of problems using function overloading and default arguments
- Develop solutions for a range of problems using access specifiers, objects and classes.
- Write, compile and run the C++ programs based on dynamic memory management methods, constructors, destructors and operator overloading.
- Describe and interpret fundamental algorithmic problems including type casting, inheritance, and polymorphism.
- Apply generic programming, templates, and file handling methods to make the code more maintainable.

Unit I

1. Write a C++ program to display the message “Hello World”.
2. Write a C++ program to print personal information using the output operator (cout).
3. Write a C++ program to add two numbers using user input (cin).
4. Write a C++ program to perform subtraction, multiplication, and division on three numbers.
5. Write a C++ program to determine whether a number is even or odd.
6. Write a C++ program to check whether a number is positive or negative, and compute the factorial if it is positive.
7. Write a C++ program to display elements of a static array.
8. Write a C++ program to input and display elements of an array using user input.
9. Write a C++ program to determine whether an entered character is a vowel or consonant.
10. Write a C++ program to demonstrate function overloading for calculating the area of different shapes.
11. Write a C++ program to demonstrate function overloading for calculating the perimeter and addition of numbers.
12. Write a C++ program to demonstrate the use of default arguments in function overloading.
13. Write a C++ program to compute Simple Interest using default arguments.
14. Write a C++ program that demonstrates returning a reference variable from a function.
15. Write a C++ program that returns a reference to an array element using a function.
16. Write a C++ program to calculate the power of a number.
17. Write a C++ program to demonstrate the use of the scope resolution operator.
18. Write a C++ program to conduct an election where five candidates receive votes. Count valid votes and count spoiled ballots using arrays.
19. Write a C++ program to create a Student class to store roll number, name, and marks of five subjects.
20. Write a C++ program to create an Employee class to input and display employee details.

21. Write a C++ program using a class to find the largest among three numbers.

22. Write a C++ program using a class to display a date in dd/mm/yyyy format.

23. Write a C++ program using classes and objects to compute the factorial of a number.

24. Write a C++ program to call member functions using a pointer to object and pointer to member function.

25. Write a C++ program to illustrate the use of the this pointer.

Unit 2

26. Write a C++ program to access private data members of two different classes using a friend function.

27. Write a C++ program to compute the average of values using a friend function.

28. Write a C++ program to compute average using a friend function with user input.

29. Write a C++ program to demonstrate a parameterized constructor for student fee details.

30. Write a C++ program to demonstrate a parameterized constructor using a Car class example.

31. Write a C++ program using a friend function to find the maximum of two numbers from two different classes.

32. Write a C++ program to create a Currency class with rupees and paisa, and implement a friend function AddCurrency() to add two currency objects.

33. Write a C++ program to create a Calendar class with day, month, and year, including constructors and a function AddDays().

34. Write a C++ program to demonstrate a parameterized dynamic constructor.

35. Write a C++ program to demonstrate a copy constructor.

36. Write a C++ program to create a String class using default, parameterized, and copy constructors.

Unit 3

37. Write a C++ program illustrating the basic concept of base and derived classes.

38. Write a C++ program to create a student report card using inheritance.

39. Write a C++ program to design a mobile scheme billing system, using Scheme (base) and Customer (derived) classes.

40. Write a C++ program to build a car showroom application using inheritance between Car and Model classes.

41. Write a C++ program to demonstrate inheritance for Publisher, Book, and Tape classes.

42. Write a C++ program to compute weekly pay for Male and Female Employees using function overriding.

43. Write a C++ program to implement Account, Current Account, and Savings Account classes with deposit, withdrawal, and interest features.

44. Write a C++ program to demonstrate function overriding.

45. Write a C++ program to demonstrate the concept of virtual functions.

46. Write a C++ program using virtual functions for Vehicle, Scooter, and Car classes.

47. Write a C++ program using a base class Shape and derived classes Triangle and Rectangle with a virtual function display_area().

48. Write a C++ program to demonstrate the use of a pure virtual function (abstract class).

49. Write a C++ program to demonstrate constructor and destructor invocation in multiple inheritance.

UNIT IV

50. Write a C++ program to overload the + operator to concatenate two strings.

51. Write a C++ program to overload the ==, <, and > operators for a Distance class containing feet and inches.

52. Write a C++ program to overload the + and − operators for a Matrix class.

53. Write a C++ program to overload prefix and postfix increment and decrement operators for a Coord class.

54. Write a C++ program to perform type conversion between Rupees and Paise classes.

55. Write a C++ program to implement Celsius ↔ Fahrenheit conversion using type conversion mechanisms.

56. Write a function template to find the maximum value in an array.

57. Write a class template to implement a generic Array class.

58. Write a function template to perform Bubble Sort.

59. Write a C++ program to demonstrate input/output using insertion and extraction operators in text mode.

60. Write a C++ program to demonstrate the use of put(), get(), and getline() functions.

61. Write a C++ program to demonstrate binary file I/O using read() and write().

62. Write a C++ program to demonstrate the use of file manipulators.

63. Write a C++ program to demonstrate file pointer manipulation functions.

64. Write a C++ program to copy the contents of one file (source.txt) into another file**.**