

SEMESTER-II

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

BSC IT Category – IV

Major Course – 1: Principles of Object-Oriented Programming

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		
Principles of Object Oriented Programming ITMC221C	2	0	2		

Learning Objectives:

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

- The foundational concepts of Object-Oriented Programming (OOP) and differentiate it from Procedural Programming.
- The syntax and usage of C++ constructs like data types, keywords, and header files.
- The importance and application of function overloading, default arguments, and inline functions in C++.
- The concepts of classes, objects, access specifiers, and the 'this' pointer for effective data encapsulation.
- The principles of inheritance, virtual functions, and polymorphism for code reusability and flexibility.
- The advanced concepts like operator overloading, templates, and dynamic memory management in C++.

Learning Outcomes:

Upon successful completion, students will be able to implement:

- Clearly differentiate between procedure-oriented and object-oriented programming approaches.
- Demonstrate a theoretical understanding of C++ syntax, including keywords, header files, and reference variables.
- Analyze and explain the role of constructors, destructors, and memory management techniques.
- Articulate the concept of inheritance, its types, and the use of virtual functions for runtime polymorphism.
- Explain operator overloading, its rules, and its applications in enhancing the readability of C++ programs.
- Discuss the benefits and limitations of templates for achieving code reusability and type safety.

UNIT	Particulars	Hours
1	Overview of Object-Oriented Programming ▪ 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 ▪ Reference Variables in C++ ▪ The bool Data type ▪ Importance of function prototyping in C++ ▪ Function Overloading ▪ Default Arguments ▪ Inline Function ▪ Scope Resolution Operator	15
	Classes and Object ▪ 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	
	More on Classes and Objects ▪ 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

2

- Writing basic C++ programs with loops, conditions and functions
- Creating Simple classes and objects along with access specifiers
- Implementing
 - Friend Function and Friend Class,
 - DMM
 - constructors and its types
 - destructors.

30

Inheritance and Virtual Functions

- 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
- 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

3

Operator Overloading

15

- 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
- Practical implementation on:
 - Inheritance and its types
 - Virtual functions
 - Operator Overloading
 - Constructor and destructor in inheritance
 - Templates

Textbook:

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

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

Sample Programs for Object-Oriented Programming with C++

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.
7. Write a C++ program to demonstrate function overloading for calculating the area of different shapes.
8. Write a C++ program to demonstrate function overloading for calculating the perimeter and addition of numbers.
9. Write a C++ program to demonstrate the use of default arguments in function overloading.
10. Write a C++ program to compute Simple Interest using default arguments.
11. Write a C++ program that demonstrates returning a reference variable from a function.

12. Write a C++ program that returns a reference to an array element using a function.
13. Write a C++ program to calculate the power of a number.
14. Write a C++ program to demonstrate the use of the scope resolution operator.
15. Write a C++ program to input and display elements of an array using user input.
16. Write a C++ program to create a student class to store roll number, name and marks of five subjects.
17. Write a C++ program to create an employee class to input and display employee details.
18. Write a C++ program using a class to find the largest among three numbers.
19. Write a C++ program using classes and objects to compute the factorial of a number.
20. Write a C++ program to call member functions using a pointer to object and pointer to member function.
21. Write a C++ program to illustrate the use of this pointer.

Unit II

22. Write a C++ program to access private data members of two different classes using a friend function.
23. Write a C++ program to compute the average of values using a friend function.
24. Write a C++ program to compute average using a friend function with user input.
25. Write a C++ program to demonstrate a parameterized constructor for student fee details.
26. Write a C++ program to demonstrate a parameterized constructor using a Car class example.
27. Write a C++ program using a friend function to find the maximum of two numbers from two different classes.
28. Write a C++ program to create a Currency class with rupees and paisa and implement a friend function AddCurrency() to add two currency objects.
29. Write a C++ program to demonstrate a parameterized dynamic constructor.
30. Write a C++ program to demonstrate a copy constructor.
31. Write a C++ program to create a String class using default, parameterized and copy constructors.

Unit III

32. Write a C++ program illustrating the basic concept of base and derived classes.
33. Write a C++ program to create a student report card using inheritance.
34. Write a C++ program to build a car showroom application using inheritance between Car and Model classes.
35. Write a C++ program to compute weekly pay for Male and Female Employees using function overriding.
36. Write a C++ program to demonstrate function overriding.
37. Write a C++ program to demonstrate the concept of virtual functions.
38. Write a C++ program using virtual functions for Vehicle, Scooter and Car classes.
39. Write a C++ program using a base class Shape and derived classes Triangle and Rectangle with a virtual function display_area().
40. Write a C++ program to demonstrate the use of a pure virtual function (abstract class).
41. Write a C++ program to demonstrate constructor and destructor invocation in multiple inheritance.

Unit IV

42. Write a C++ program to overload the + operator to concatenate two strings.
43. Write a C++ program to overload the ==, < and > operators for a Distance class containing feet and inches.
44. Write a C++ program to perform type conversion between Rupees and Paise classes.
45. Write a C++ program to implement Celsius to Fahrenheit conversion using type conversion mechanisms.
46. Write a C++ program to demonstrate the use of function template.
47. Write a C++ program to demonstrate the use of class template.

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

BSc-IT (Hons.) Category – IV

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		
Introduction to Cybersecurity ITMC222C	4	0	0	Nil	Programming Basics

Learning Objectives:

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

The objective of the course is to understand the fundamentals of cryptography, various encryption mechanisms, different key distribution techniques and deployment of encryption schemes to secure data across data networks

Learning Outcomes:

Upon successful completion, students will be able to implement:

- Explain the cybercrime terminologies
- Describe Cyber offenses and Botnets
- Illustrate Tools and Methods used on Cybercrime
- Explain Phishing \ and Identity Theft
- Justify the need of computer forensics

UNIT	Particulars	Hours
1	Introduction to Cybercrime: Definition and Origins of the Word: cyberspace, cybersquatting, cyberpunk and cyberwarfare and cyberterrorism, Cybercrime and Information Security: How Zombie works?, Who are Cybercriminals? Classifications of Cybercrimes, Cybercrime An Indian Perspective, Hacking and Indian Laws., Global Perspectives, Cybercrime Era: Survival mantra for Netizens	15
2	Cyber Offenses: How Criminals Plan Them: Introduction: Hacker, Crackers, Phreakers, What colour s your hat in the security	30

world?, categories of cybercrime, How criminals plan the attacks, Social Engineering, Cyber Stalking: Type of stalkers, How stalking works, Cyberbullying, Case study on real-life incident of cyberstalking, Cybercaafe & cybercrimes: safety measures.

Botnets:

The fuel for cybercrime, Attack Vector.

- | | | |
|--|---|----|
| 3 | Tools and Methods used in Cybercrime:
Introduction, Proxy Servers, Anonymizers, Phishing, Password Cracking, Key Loggers and Spyways, Virus and Worms, Trozen Horses and Backdoors, Steganography, DoS and DDOS Attackers, Attacks on Wireless networks | 15 |
|
Phishing and Identity Theft:
Introduction, methods of phishing | | |
| 4 | Understnading Computer Forensics:
Introdcution, Historical Background of Cyberforensics, Digital Foresics Science, Need for Computer Foresics, Cyber Forensics and Digital Evidence, Digital Forensic Life cycle, Chain of Custody Concepts, network forensics. | 30 |

Textbook:

William Stallings: Network Essentials: applications and standards, Pearson 6th edition

Reference Books:

1. William Stallings: Network Essentials: applications and standards, Pearson 4th edition
 2. William Stallings: Cryptography and Network Security, Pearson 6th edition
- V k Pachghare: Cryptography and Information Security. PHI Learning. ISBN 978- 81-203-3521-9