C. U. SHAH UNIVERSITY
Wadhwan City

FACULTY OF: Computer Science
DEPARTMENT OF: Bachelor of Computer Application
SEMESTER : II
CODE: 4CS02BOP1
NAME: Object Oriented Programming with C++

Teaching and Evaluation Scheme

Sr. No	Subject Code	Subject Name	Teaching Hours/Week				Credits	Evaluation Scheme/Semester							
								Theory				Practical			Total Marks
								Sessional Exam		University Exam		Internal		Uni.	
			Th	Tu	Pr	Total		Marks	Hrs	Marks	Hrs	Pr	TW	Pr	
2	4CS02BOP1	Object Oriented Programming with C++	4	-	-	4	4	30	1.5	70	3	-	-	-	100

Objectives:

- 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.

Pre-requisites: Fundamental knowledge of C Programming

Course Outline:

Ch. No	Chapter Name	Content	Lect. Hours
1	**OOPS Introduction**	**1.1 Overview of Object Oriented Programming** 1.1.1 Introduction to Object Oriented Programming 1.1.2 Procedure Oriented and Object Oriented 1.1.3 Difference Between C and C++ 1.1.4 C++ Output/ Input 1.1.5 Keywords in C++ 1.1.6 New style of header file specification 1.1.7 Comments in C++ 1.1.8 Variables in C++ 1.1.9 Reference Variables in C++ 1.1.10 The bool Data type 1.1.11 Importance of function prototyping in C++ 1.1.12 Function Overloading 1.1.13 Default Arguments	10

		1.1.14 Inline Function 1.1.15 Scope Resolution Operator **1.2 Class and object** 1.2.1 Structures in C 1.2.2 Structure in C++ 1.2.3 Access Specifier 1.2.4 Classes 1.2.5 Objects in C++ 1.2.6 Characteristics of Access Specifier 1.2.7 Function outside a class 1.2.8 Initialization of variable in C++ 1.2.9 Arrow Operator 1.2.10 'this' pointer	
2	**More on C++ Classes and Object, Dynamic Memory Management, Constructor & Destructor**	**2.1 More on Classes and Objects** 2.1.1 Member Functions and Data Members 2.1.2 Friend Functions 2.1.3 Friend Class 2.1.4 Array of Class Object 2.1.5 Passing Class Objects to Function 2.1.6 Returning Objects from Functions 2.1.7 Nested Classes 2.1.8 Namespaces **2.2 Dynamic Memory Management** 2.2.1 Introduction 2.2.2 Dynamic Memory Allocation Using "new" 2.2.3 Dynamic Memory Deallocation 2.2.4 "Set_New_Handler" Function **2.3 Constructor and Destructor** 2.3.1 Constructor 2.3.2 Characteristics of Constructor 2.3.3 Types of Constructor 2.3.4 Destructor 2.3.5 Characteristics of Destructor	15
3	**Inheritance and Polymorphism**	**3.1 Inheritance** 3.1.1 Introduction 3.1.2 Advantages of Inheritance 3.1.3 'Protected' Access specifier 3.1.4 Inheritance using different access specifier 3.1.5 Initialization of Base class members through derived class object 3.1.6 Different forms of Inheritance 3.1.7 Function Overriding **3.2 Virtual function and inheritance** 3.2.1 Introduction 3.2.2 Pointers to derived class	15

		3.2.3 Rules for virtual function 3.2.4 Internals of Virtual Functions 3.2.5 Pure virtual function 3.2.6 Virtual Base class 3.2.7 Virtual destructor 3.2.8 Abstract class 3.2.9 Limitations of virtual Function 3.2.10 Early binding v /s Late binding	
4	**Operator Overloading, Constructor-Destructor Invocation and Templates**	**4.1 Operator Overloading** 4.1.1 Introduction 4.1.2 Operators that can be overloaded 4.1.3 Overloading Unary Operator using member Functions 4.1.4 Overloading Unary Operator using friend Functions 4.1.5 Overloading Binary Operator using member Functions 4.1.6 Overloading Binary Operator using friend Functions 4.1.7 Why to Overload Operators using friend Function? 4.1.8 Rules for Operator Overloading **4.2 Constructor- Destructor Invocation** 4.2.1 Introduction 4.2.2 Order of Invocation of Constructors and destructors 4.2.3 Destructors in Action 4.2.4 Type Conversions **4.3 Templates** 4.3.1 Introduction 4.3.2 Function Templates 4.3.3 Function Templates with multiple parameters 4.3.4 Overloading Function Template 4.3.5 Class Template 4.3.6 Class Template with multiple parameters 4.3.7 Nested Class Templates o Advantages of using Templates	15
		Total	**55**

Text Books			
SR No.	**Book Name**	**Author**	**Publication**
1	Object Oriented Programming with C++	Subhash KU	Pearson
Reference Books			
SR No.	**Book Name**	**Author**	**Publication**
1	Object-Oriented Programming with C++ (Second Edition)	Poornachandra Sarang	PHI
2	Object Oriented Programming using C++	Joyce Farrell	Cengage Learning
3	Object Oriented Programming In C++	Rajesh K. Shukla	Wiley India Edition

Program list

Chapter 1 : OOPS Introduction

1. Write a program to calculate the area of circle, rectangle and square using function overloading.
2. Write a program to demonstrate the use of default arguments in function overloading.
3. Write a program to demonstrate the use of returning a reference variable.
4. Create a class student which stores the detail about roll no, name, marks of 5 subjects, i.e. science, Mathematics, English, C, C++. The class must have the following:

   • Get function to accept value of the data members.

   • Display function to display values of data members.

   • Total function to add marks of all 5 subjects and store it in the data members named total.

5. Create a function power() to raise a number m to power n. the function takes a double value for m and int value for n, and returns the result correctly. Use the default value of 2 for n to make the function calculate squares when this argument is omitted. Write a main that gets the values of m and n from the user to test the function.
6. Write a basic program which shows the use of scope resolution operator.
7. Write a C++ program to swap the value of private data members from 2 different classes.
8. Write a program to illustrate the use of this pointer.
9. An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the votes cast for each candidate using an array variable count. In case a number is read outside the range of 1 to 5, the ballot should be considered as a 'spoilt ballot' and the program should also count the number of spoilt ballots.
10. Write a program to call member functions of class in the main function using pointer to object and pointer to member function.

Chapter 2 : More on C++ Classes and Object, Dynamic Memory Management, Constructor & Destructor

11. Using friend function find the maximum number from given two numbers from two different classes. Write all necessary functions and constructors for the program.
12. Using a friend function, find the average of three numbers from three different classes. Write all necessary member functions and constructor for the classes.
13. Define currency class which contains rupees and paisa as data members. Write a friend function named AddCurrency ( ) which add 2 different Currency objects and returns a

Currency object. Write parameterized constructor to initialize the values and use appropriate functions to get the details from the user and display it.

14. Create Calendar class with day, month and year as data members. Include default and parameterized constructors to initialize a Calendar object with a valid date value. Define a function AddDays to add days to the Calendar object. Define a display function to show data in "dd/mm/yyyy" format.
15. Create a class named 'String' with one data member of type char *, which stores a string. Include default, parameterized and copy constructor to initialize the data member. Write a program to test this class.

Chapter : 3 Inheritance and Polymorphism

16. Write a base class named Employee and derive classes Male employee and Female Employee from it. Every employee has an id, name and a scale of salary. Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of days and hours he works. The female employee gets paid the wages for 40 hours a week, no matter what the actual hours are. Test this program to calculate the pay of employee.
17. Create a class called scheme with scheme_id, scheme_name, outgoing_rate, and message_charge. Derive customer class form scheme and include cust_id, name and mobile_no data. Define necessary functions to read and display data. Create a menu driven program to read call and message information for a customer and display the detail bill.
18. Write a program with use of inheritance: Define a class publisher that stores the name of the title. Derive two classes book and tape, which inherit publisher. Book class contains member data called page no and tape class contain time for playing. Define functions in the appropriate classes to get and print the details.
19. Create a class account that stores customer name, account no, types of account. From this derive classes cur_acc and sav_acc to include necessary member function to do the following:

    • Accepts deposit from customer and update balance

    • Compute and Deposit interest

    • Permit withdrawal and Update balance.

20. Write a base class named Employee and derive classes Male employee and Female Employee from it. Every employee has an id, name and a scale of salary. Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of days and hours he works. The female employee gets

paid the wages for 40 hours a week, no matter what the actual hoursare. Test this program to calculate the pay of employee.

21. Create a class vehicle which stores the vehicleno and chassisno as a member. Define another class for scooter, which inherits the data members of the class vehicle and has a data member for a storing wheels and company. Define another class for which inherits the data member of the class vehicle and has a data member for storing price and company. Display the data from derived class. Use virtual function.
22. Create a base class shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize the base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived class to suit their requirements.
23. Write a program to demonstrate the use of pure virtual function.
24. For multiple inheritance, write a program to show theinvocation of constructor and destructor.

Chapter 4: Operator Overloading, Constructor-Destructor Invocation and Templates

25. Create a class string with character array as a data member and write a program to add two strings with use of operator overloading concept.
26. Create a class distance which contains feet and inch as a data member. Overhead = =, <and> operator for the same class. Create necessary functions and constructors too.
27. Create a class MARIX of size mxn. Overload + and – operators for addition and subtraction of the MATRIX.
28. Define a class Coord, which has x and y coordinates as its data members. Overload ++ and – operators for the Coord class. Create both its prefix and postfix forms.
29. Create one class called Rupees, which has one member data to store amount in rupee and create another class called Paise which has member data to store amount in paise. Write a program to convert one amount to another amount with use of type conversion.
30. Create two classes Celsius and Fahrenheit to store temperature in terms of Celsius and Fahrenheit respectively. Include necessary functions to read and display the values. Define conversion mechanism to convert Celsius object to Fahrenheit object and vice versa. Show both types of conversions in main function.
31. Write a program to create a function template for finding maximum value contained in an array.
32. Write a program to create a class template for the ‘Array’ class.
33. Create a template for the bubble sort function.