



GUJARAT UNIVERSITY

BCA III SYLLABUS

COURSE TITLE	Object Oriented Concepts and Programming
COURSE CODE	CC-203
COURSE CREDIT	3
Session Per Week	4
Total Teaching Hours	40 HOURS

AIM

- 1.) To get in-depth knowledge of Object Oriented Programming language.
- 2.) To obtain knowledge of programming for real life applications.

LEARNING OUTCOMES

1. Understand the features of C++ supporting object oriented programming
2. Understand the relative merits of C++ as an object oriented programming language
3. Understand how to produce object-oriented software using C++
4. Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism.
5. Understand advanced features of C++ specifically stream I/O, templates and operator overloading

DETAIL SYLLABUS

UNIT	TOPIC / SUB TOPIC	TEACHING HOURS
	OOPS Introduction	10
1	* Overview of Object Oriented Programming - o Introduction to Object Oriented Programming - o Procedure Oriented and Object Oriented - o Difference Between C and C++ - o C++ Output/ Input - o Keywords in C++ - o New style of header file specification - o Comments in C++ - o Variables in C++ - o Reference Variables in C++ - o The bool Data type - o Importance of function prototyping in C++ - o Function Overloading - o Default Arguments - o Inline Function - o Scope Resolution Operator	6

	Classes And Object - o Structures in C - o Structure in C++ - o Access Specifier - o Classes - o Objects in C++ - o Characteristics of Access Specifier - o Function outside a class - o Initialization of variable in C++ - o Arrow Operator - o 'this' pointer	4
2	More on++Classes and Object, Dynamic Memory Management, Constructor & Destructor	10
	* More on Classes and Objects - o Member Functions and Data Members - o Friend Functions - o Friend Class - o Array of Class Object - o Passing Class Objects to Function - o Returning Objects from Functions - o Nested Classes - o Namespaces	5
	* Dynamic Memory Management - o Introduction - o Dynamic Memory Allocation Using "new" - o Dynamic Memory Deallocation	2
	* Constructor and Destructor - o Constructor - o Characteristics of Constructor - o Types of Constructor - o Destructor - o Characteristics of Destructor	3
	Inheritance and Polymorphism	10
	* Inheritance - o Introduction - o Advantages of Inheritance - o 'Protected' Access specifier - o Inheritance using different access specifier - o Initialization of Base class members through derived class object - o Different forms of Inheritance - o Function Overriding	5

3	* Virtual Functions and Inheritance - o Introduction - o Pointers to derived class - o Rules for virtual function - o Internals of Virtual Functions - o Pure virtual function - o Virtual Base class - o Virtual destructor - o Abstract class - o Limitations of virtual Function - o Early binding v /s Late binding	5
4	Operator Overloading, Working with Files and Templates	10
	* Operator Overloading - o Introduction - o Operators that can be overloaded - o Overloading Unary Operator using member Functions (-,++ and --) - o Overloading Unary Operator using friend Functions (+,-,++ and --) - o Overloading Binary Operator using member Functions (+,-,*,/,>,<==,!=,>= and <=) - o Overloading Binary Operator using friend Functions (+,-,*,and /) - o Why to Overload Operators using friend function? - o Rules for Operator Overloading - o Type Conversions * basic type to class type * class type to basic type * class type to another class type - o Excluding Assignment operators (=,+=,*=,/=,-=,%=,&=, =,^=), Bit-wise Operator, Dereferencing, New, Delete, Subscript, Function call, Logical and >>=,<<=	5
	* Working with Input, Output and Files - o Introduction - o Stream Class Model of C++ (istream,ostream,ifstream,ofstream,iostream) - o Text Files - o Text mode input using 'extraction'(>>) operator, 'get()' function and 'getline()' function - o Text mode output using 'insertion' (<<) operator and 'put()' function	2
	* Templates - o Introduction - o Function Templates - o Function Templates with multiple parameters - o Overloading Function Template - o Class Template - o Class Template with multiple parameters - o Nested Class Templates - o Advantages of using Templates	3
TEXT BOOK/S:		



GUJARAT UNIVERSITY

BCA III SYLLABUS

COURSE TITLE	C++ Practicals
COURSE CODE	CC-207
COURSE CREDIT	3
Session Per Week	3
Total Teaching Hours	40 HOURS

AIM

- 1.) To get in-depth practical knowledge of Object Oriented Programming language.
- 2.) To obtain practical knowledge of programming for real life applications.

LEARNING OUTCOMES

1. Understand the features of C++ supporting object oriented programming
2. Understand the relative merits of C++ as an object oriented programming language
3. Understand how to produce object-oriented software using C++
4. Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism.
5. Understand advanced features of C++ specifically stream I/O, templates and operator overloading

DETAIL SYLLABUS

UNIT	TOPIC / SUB TOPIC	TEACHING HOURS
	Introduction to OOP, Classes & Objects	10
	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 Storeit in the data members named total.	

1	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.	10
2	Dynamic Memory Management, Constructor & Destructor, Inheritance 1. Using friend function find the maximum number from given two numbers from two different classes. Write all necessary functions and constructors for the program. 2. Using a friend function, find the average of three numbers from three different classes. Write all necessary member functions and constructor for the classes. 3. 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. 4. 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. 5. 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. 6. 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.	10

7. Create a class called scheme with scheme_id, scheme_name, outgoing_rate, and message_charge. Derive customer class from 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.

8. 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 contains time for playing. Define functions in the appropriate classes to get and print the details.

9. 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.

10. 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

Virtual Functions, Operator Overloading

10

1. Create a class vehicle which stores the vehicle no and chassis no as a member. Define another class for scooter, which inherits the data members of the class vehicle and has a data member for 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.

2. 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.

3	3 Write a program to demonstrate the use of pure virtual function. 4 Create a class time with member data hour and minute. Overload ++ unary operator for class time for increment and -- unary operator for decrement in time object value. 5 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. 6 Create a class distance which contains feet and inch as a datamember. Overhead = =, <and> operator for the same class. Create necessary functions and constructors too. 7 Create a class MATRIX of size mxn. Overload + and – operators for addition and subtraction of the MATRIX. 8 Define a class Coord, which has x and y coordinates as itsdata members. Overload ++ and – operators for the Coordclass. Create both its prefix and postfix forms 9 Create one class called Rupees, which has one member data tostore 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. 10 Create two classes Celsius and Fahrenheit to store temperaturein terms of Celsius and Fahrenheit respectively. Includenecessary functions to read and display the values. Defineconversion mechanism to convert Celsius object to Fahrenheitobject and vice versa. Show both types of conversions in mainfunction.	
4	Templates, Files 1 Write a program to create a function template for finding maximum value contained in an array. 2 Write a program to create a class template for the 'Array' class. 3 Create a template for the bubble sort function. 4 Write a program to create a function template for swapping the two value. 5 Write a program to illustrate the use of put(), get() and getline() functions for Text mode Input/Output. 6 Write a program to read character, integer and string from keyboard and write it in "data.txt" file and read from file in text mode. 7 Write a program to read your name and roll number from keyboard and write it in "mydata.txt " file and read from file in text mode. 8 Write a program to read product name and product price from keyboard and write it in "product.txt" file and read from file in text mode. 9 Write down a program to create a file temp.txt, write into the specific file than read the same data from the file 10 Write a program to create num.txt file which stores number. Find max value from a file nums.txt and print it on standard output device.	10

TEXT BOOK/S:

1. 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

WEB RESOURCES:

REQUIRED SOFTWARE/S

Turbo C