BVM ENGINEERING COLLEGE [AN AUTONOMOUS INSTITUTION]

2EC07: OBJECT ORIENTED PROGRAMMING LABORATORY
CREDITS - 1 (LTP:0,0,1)

Course Objective:
To learn the programming of basics of C++, Objects and Classes, Inheritance, Polymorphism, I/O and file management, and advance topics including templates, exceptions and STL (Standard Template Library).

Teaching and Assessment Scheme:

Teaching Scheme (Hours per week)			Credits	Assessment Scheme				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE	CE	ESE	CE	100
0	0	2	1	00	00	40	60	

List of Experiments:

Unit No.	SET-1 Basic C++ Programs
1.	How to Compile and Run C++ program in Ubuntu Linux?
2.	Write a program to print hello world on the screen.
3.	Write a program to read integer, float and string from the keyboard and display it on the screen.
4.	Write a program to find average of two numbers.
5.	Write a program to check whether number is even or odd.
6.	Write a program to make a simple calculator.
7.	Write a program that uses structure to store three parts of phone number separately.
8.	Write a program to implement concept of reference variable.
9.	Write a program to find sum of digits of a number.
10.	Write a program for function with default arguments to find the area of circle.
11	Write a program to illustrate the concept of call by value and call by reference (swapping of two numbers).
12.	Write a program to use the concept of scope resolution operator
13.	Write a program to implement concept of function overloading (area calculation).
14.	Write a program to implement function overloading to computer pow (m,n). Case (1) m is integer and n is double, (2) m and n both are int .

Unit No.	SET-2 Programs using Class and Objects
15.	Write a program to demonstrate declaration and initialization of object of class. (to print students age, name and sex).
16.	Write a program to demonstrate declaration and initialization of object of class. (Member function inside the class)
17.	Write a program to find smallest number out of two classes with help of class and object.
18.	Write a program to perform operation on string class (find length, compare two strings and join them).
19.	Write a program to find whether number is Armstrong or not using class and object.
20.	Write a program to implement concept of manipulator setw ().

21.	Write a program to implement concept of explicit type casting.
22.	Write a program to find cubic values using inline function.
23.	Write a program to find factorial of given number (function recursion).
24.	Write a program to implement concept of math functions.
25.	Write a program to find the largest value out of two numbers using nesting of member functions.
26.	Write a program to implement nesting of member functions using class to find the 1's complement of given number.
27.	Write a program to process a shopping list.
28.	Write a program to implement concept of static class member.
29.	Write a program to implement concept of static member functions
30.	Write a program to implement concept of array of object.
31.	Write a program for addition of two time objects (objects as arguments) .
32.	Write a program to find mean value of given number using friend function.
33.	Write a program to add data objects of two different classes using friend functions
34.	Write a program to add two complex numbers using concept of returning objects.
35.	Write a program to implement friend function to two classes.
36.	Write a program to swap private data of two classes.
37.	Write a program to transpose 3x3 matrices by concept of returning objects.
38.	Write a program to add two numbers with the help of dereferencing operator.
39.	Write a program to implement concept of banking system.

Unit No.	**SET-3 Programs using Constructor and Destructor**
40.	Write a program to implement concept of constructor and destructor.
41.	Write a program to check whether number is prime or not.
42.	Write a program to implement the concept of copy constructor.
43.	Write a program to calculate factorial of a given number using copy constructor.
44.	Write a program to implement concept of overloaded constructor.
45.	Write a program for dynamic initialization of constructor.
46.	Write a program to construct strings in objects using dynamic constructor.

Unit No.	**SET-4 Programs using Operator Overloading**
47.	Write a program to implement concept of unary operator overloading.
48.	Write a program to implement concept of unary operator overloading using friend function.
49.	Write a program to implement concept of binary operator overloading.
50.	Write a program to implement concept of binary operator overloading using friend function.
51.	Write a program for multiplication of vector using operator overloading with friend function.
52.	Write a program to perform mathematical operations on strings.
53.	Write a program to add and multiply to matrix objects by overloading + and * operators using friend functions
54.	Write a program to convert data of one type to another using two classes.

Unit No.	**SET-5 Programs using Inheritance**
55.	Write a program to implement concept of single inheritance using public class.
56.	Write a program to implement concept of single inheritance using private class.
57.	Write a program to implement concept of multilevel inheritance.
58.	Write a program to implement concept of multiple inheritance.

59.	Write a program to implement concept of Hybrid inheritance.
60.	Write a program to implement concept of virtual base class using inheritance.
61.	Write a program to implement concept of constructor in derived class using inheritance.
62.	Write a program to implement concept of initialization list in constructors using inheritance.

Unit No.	SET-6 Programs using Pointers, Virtual Functions and Polymorphism
63.	Write a program to implement concept of manipulation of pointers using dereference operator.
64.	Write a program to perform arithmetic operations on pointers.
65.	Write a program to implement concept of pointers with arrays.
66.	Write a program to implement concept of array of pointer.
67.	Write a program to implement concept of pointers to function.
68.	Write a program to implement concept of pointers to objects.
69.	Write a program to implement concept of array of pointer to objects.
70.	Write a program to implement concept of 'this' pointer.
71.	Write a program to implement concept of pointers to derived objects.
72.	Write a program to implement concept of virtual functions.
73.	Write a program to implement concept of run time polymorphism using virtual function.

Unit No.	SET-7 Programs using I/O Operation, Files, Template and Exceptional Handling
74.	Write a program that uses a single file for both reading and writing the data.
75.	Write a program for specifying field size using width ().
76.	Write a program for precision setting using precision ().
77.	Write a program for padding with fill ().
78.	Write a program to design your own manipulator to provide the following output specification for printing money value.
79.	Write a program to create a database regarding the people admit in a hospital.
80.	Write a program to swap a number using concept of function template.
81.	Write a program to create a class template to find the greater of the two numbers
82.	Write a program to perform exception handling operation.
83.	Write a program to perform exception handling for divide by zero.

List of References:

1. E Balagurusamy, *"Object Oriented Programming with C++"*, Tata McGraw Hill, 6th edition, 2013.
2. Bhushan Trivedi, *"Programming with ANSI C++"*, Oxford University Press, 2nd edition, 2013.
3. Herbert Schlitdt, *"C++ the Compete Reference"*, TMH Publication, 5th edition, 2014.
4. Ashok Kamthane, *"Object Oriented Programming with ANSI and Turbo C++"*, Pearson Education, 1st Edition, 2011.

Course Outcomes (COs):

By learning this course students will be able to …

1. Remember the basic programming concepts and the necessary constructs of C++ programming.
2. Understand algorithmic thinking and problem solving techniques.
3. Describe concepts like object and class, encapsulation, inheritance, polymorphism, template and exception handling for real life problems.
4. Apply advance features to make programs which supports reusability and sophistication.
5. Test standard template library for faster development.
6. Resolve C++ program using variables, operators, control structures, functions and objects.