AMITY UNIVERSITY

JHARKHAND

LAB MANUAL

Course Title: Object Oriented Programing Using C++

Course Level: UG

Course Code: ES203

Program: B. Tech. (CE+ECE+MAE+CSE)

Semester: III

Faculty:

MS. TANNISHA KUNDU
Assistant Professor (CS)

Approved By:

PROF. (DR.) AJIT KUMAR PANDEY
Director, AUJ

Lab Instructor:
MR. RAHUL KUMAR LOHRA

General instructions to students

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual.
4. Prescribed textbook and class notes can be kept ready for reference if required.
5. They should implement the given experiment individually.
6. Once the experiment(s) get executed, they should show the program and results to the instructors and copy the same in their observation book.
7. Questions for lab tests and exam need not necessarily be limited to the questions in the manual but could involve some variations and / or combinations of the questions.
8. All the students must maintain silence inside the lab.
9. All the students must carry their id card before entering the lab, and college uniform is strictly mandatory, otherwise students will not be permitted to sit inside the lab.
10. No food or beverage items are allowed inside the lab.
11. Keep your bags outside the lab.
12. Do not use cell phone inside the lab. (If anybody found using cell phone inside the lab, his/her mobile phone will be seized by the responsible authority.)
13. Shut down the system and arrange your chair before leaving the lab.
14. While using lab sign in the register with your name, enrollment number and branch to mark your attendance.
15. Do not plug any device without permission.
16. Do not use internet without permission.

Note: Above mentioned instructions can be modified based on the context of the lab.

Credit Unit

L (Lecture)	T	P/S (Practical/Studio)	SW/FW	Total Credit Units
3	-	2	-	4

Lab/ Practical/ Studio Assessment

	Continuous Assessment/Internal Assessment				End Term Examination
Components (Drop down)	**Performance**	**Viva**	**Lab Record**	**Attendance**	70
Weightage (%)	10	5	10	5	

Course Objectives:

After finishing this course student will be able to:

1) Get introduction of object-oriented programming.
2) Explore and implement the various features of oop such as inheritance, polymorphism, exceptional handling etc. Using programming language c++.
3) To easily identify the basic difference between the programming approaches like procedural and object oriented

Pre-requisites: Knowledge of Programming logic and techniques

Student Learning Outcomes:

1) To apply the knowledge gained in areas of Information Technology, Computing, Mathematics, and Basic Science & Engineering. Analyse and attract the vital resources required to turn a vision into reality.
2) To build a robust foundation in theoretical to build a robust foundation in theoretical and experimental work to analyse, create and design software products, processes and systems. Experimental work to analyse, create and design software products, processes and systems.
3) To gain expertise in designing, implementation and development of computer-based systems and IT processes.
4) To use current techniques, skills, and tools necessary for computing practice.

Pedagogy for Course Delivery:

The class will be taught using theory and case-based method. In addition to assigning the case studies, the course instructor will spend considerable time in understanding the concept of innovation through the eyes of the consumer. The instructor will cover the ways to think innovatively liberally using thinking techniques.

<u>List of Experiments</u>

Week 1: Simple C++ Programs to Implement Various Control Structures. If statement, Switch case statement and do while loop, For loop, While loop

1. Write a program in C++ to find factorial of a given number.
2. Write a program that reads numbers are entered one after another as input, terminated by a value 0. Add those numbers and display the sum.
3. Write functions for the following and show their invocation in the main.
   a) A function to search for an integer in a list of integers. If found the return value of the function is 1 else return value is -1. Pass the list of integers, length of array and the key to be searched as parameters to the function.
   b) A function which determines the biggest element in an array of integers. The function returns the number. Pass the list of integers and length of the array as parameter to the function.

Week 2: Programs to Understand Structure & Unions.

1. Define a structure to represent a student. Keep track of you, ID, Name and CGPA. Write function namely input, display and edit to input details to a student variable, to display it and to edit it respectively. Write a main function with define a structure pointer to access variables. Invoke the functions in the main and observe the result.
2. Define a structure called rectangle which keep track of the length and breadth of a rectangle. Write functions namely input, display dimensions, display area and added to input the dimension of a rectangle, to display the dimensions, to calculate and display the area of a rectangle respectively. Write a main function which define a variable of type rectangle. Invoke date functions in the main and observe the result.

Week 3: Programs to Understand Pointer Arithmetic.

1. Write a program to reverse a number and calculate the sum of the digits using pointers.
2. Write a function to swap two numbers using pointer.
3. Write a program to reverse a string and calculate the length of the string using pointers. (Do not use any string function)

Week 4: Functions, Recursion & Inline Functions.

1. Write a program to calculate the area of a triangle and a circle using inline function.
2. Write a program to display a Fibonacci series and factorial of a number using recursion.
3. Write an inline function to check whether a number is prime or not.

Week 5: Class, data members and friend functions

1. Define a class to represent a complex number. Provide the following number function: -
   i. To assign initial value to the complex object.
   ii. To display a complex number in a+ib format.
   iii. To add 2 complex numbers, download. (the return value should be complex)
   iv. To subtract 2 complex numbers. Write a main function to test the class.

2. Create a class called Time that has data members to represent hours, minutes and seconds. Provide the following member Functions: -
   i. To assign initial values to the Time object.
   ii. To display a time object in the form of hh:mm:ss { 0 to 24 hours}
   iii. To add 2 Time objects (the return value should be a Time object)
   iv. To subtract 2 time objects (the return value should be a Time object)
   v. To compare 2 time objects and to determine if the first is greater or smaller than the second one.
3. Create 2 classes namely rectangle and circle with suitable data members. Provide function namely input, display and area to these classes. Input is for inputting the dimensions, displays for displaying determination as well as area and area function is for calculating the area and displaying the same. Write a friend function of class rectangle which compares the area of the rectangle object with a circle object and displays which one is greater.

Week 6: Constructors & Destructors.

1. Consider the already defined time class. Provided default constructor, a parameterized constructor and a copy of constructor to this class. Also provided display function. Illustrate all the constructors as well as the display method by defining time objects.

2. Consider the already defined complex class. Provided default constructor, a parameterized constructor and a copy of constructor to this class. Also provided display function. Illustrate all the constructors as well as the display method by defining Complex objects.
3. Define a class to represent a bank account. Include the following members.
   a. Data member:
      i. Name of the depositor.
      ii. Account number
      iii. Type of account
      iv. Balance amount in the account.
      v. Rate of interest (static data) \

   Provide a Default constructor, a parameterized Constructor, a copy constructor and a destructor to this class. Also provide Member function -
      i. To deposit money.
      ii. To display amount after checking for minimum balance.
      iii. To display all details of an account holder.
      iv. Display rate of interest (a static function)
      v. Illustrate all the structure as well as call the method by defining object.

Week 7: Programs to Implement Inheritance and Function Overriding.

a. Multiple inheritance –Access Specifiers
b. Hierarchical inheritance – Function Overriding /Virtual Function

1. Design a base class called ***Student*** with the following 2 fields: i) Name ii) ID derive 2 classes call sports and exam from Student base class. Class Sports has a field call **s_ grade** and class Exam has a field call **e_grade** which are integer fields.
   Derive class called ***Result*** which inherit the ***Sports*** and ***Exam***. This Class has a char. array or string field to represent the final result. Also, it has a member function called display which can be used to display the final result.
   Illustrate the uses of these classes in the main.
2. Create a base class called ***Shape***. Used this class to store two double type value which could be used to compute the area of figure. Derive 2 specified class called ***Triangle*** and ***Rectangle*** from the base class ***Shape***. Add to the base class; a member function called **get_data** to initialise base class data member and another member function **display_area** to compute and display area of figure. Make **display_area** a virtual function and redefined this function in the derived classes to suit their requirement. Using these three classes design a program which will accept dimensions of a triangle or rectangle interactively and display the area.

Week 8: Programs to Overload Unary & Binary Operators as Member Function & Non-Member Function.

a. Unary operator as member function
b. Binary operator as non-member function

1. Define a class IntArr which hosts an array of integers. Provide the following member function:
   a) A default constructor.
   b) A parameterized constructor which initializes the array of the object.
   c) A copy constructor
   d) A fn. called Display to display the array contents.
   e) A fn. called search to search for an element in the array.
   f) A fn. Called compare which compares 2 IntArr objects for equality.
   g) Overload +operator to add 2 INT Arr: objects.

   h) "==" to compare for equality of 2 objects.

2. Create a class called String which consists only one data member which is character pointer. Provide the following to this class:
   a) A default constructor which initializes the pointer to null value.
   b) A parameterized constructor which receive a string as its parameter. {Note: memory allocation to be done}
   c) A copy constructor which receives a string as its parameter. {Note: memory allocation to be done}.
   d) Overload + operator to concatenate 2 string objects. {Note: proper memory allocation to be done}.
   e) A display function to display the string object.
   f) A destructor to deallocate the memory which was allocated dynamically.
   g) Change Case, which converts all lower case to upper case and vice versa.
   h) Reverse, which reverse the character array.
   i) Also overload input and Output operators.
   j) "[]" to retrieve a character from the specified index.

Week 9: Programs to Understand Friend Function & Friend Class.

   c. Friend Function
   d. Friend class

1. Write a program to demonstrate Friend Function. Design two classes A & B with and integer variable a & b. Initialize the integers of these two classes using default constructor of that class. Design a friend function called add to add these two integers and implement main.
2. Write a program to demonstrate the use of friend class. Design two class A & B, make class B a friend to Class A. Class A will have the following:
   a. 2 variables: length, breadth
   b. Function display to display length

   Class B will have the following:
   a. Function area to display the area of rectangle using length and breadth defined in class A.

Week 10: Programs on Class Templates & Function Template

1. Write a program to display largest among two numbers using function templates.
2. Write a program to swap data using function templates.
3. Write a program to add, subtract, multiply and divide two numbers using class template

****

Text Books:

- R. Lafore, "Object Oriented Programming using C++", BPB Publications, 2004
- "Object Oriented Programming with C++" By E. Balagurusamy

Web References:

- A.R. Venugopal, Rajkumar, T. Ravishanker "Mastering C++", TMH, 1997
- Schildt Herbert, "C++: The Complete Reference", Wiley DreamTech, 2005