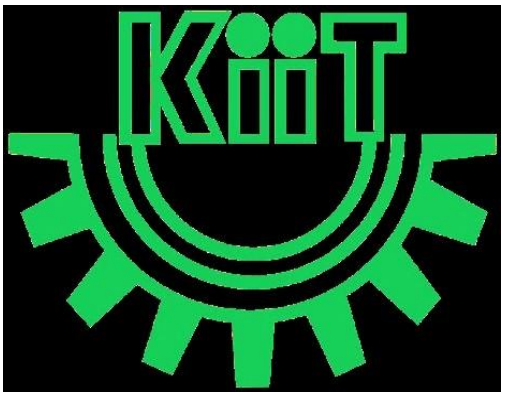
Lab Manual

Object Oriented Programming Laboratory

Code: IT-2095



Lab Coordinator
Dr.Suneeta Mohanty,
Asso. Professor, SCE
KIIT Deemed to be University

Content: Lab Manual

Content: List of Assignments

To make laboratory experiments effective, each student must obey the following rules:

1. General instructions:

- Create a directory named as your rollno_section under the home directory of UBUNTU OS system using command-line or by GUI.
- In Each lab, store programs within appropriate folders named as LAB01, LAB02, LAB03...etc. which are the sub folders under your rollno_section folder.
- Always save programs files with the meaningful name preceded by lab assignment no within specified folders. If you want solve a lab assignment no. HA3.5 (3.5 means 5th assignment of 3rd lab) which is to find roots of a quadratic equation, then name the programas HA35_quadratic.cpp or HA35_quadeq.cpp etc.

3. Laboratory Report:

At the end of every lab, student need to write the solution to the given assignments using pen and paper and need to submit in the google classroom.

4. Programming Instructions:

- The g++ compiler has to be used to compile the programs
- To Install g++, in the command prompt type "**sudo apt-get install g++** ". Then give the password.
- Save programming files with extension '.cpp'.
- Use 'gedit&' command to open the editor to write the programs.

Course objective

- Understand object oriented programming and advanced C++ concepts

    i. Be able to program using more advanced C++ features such as composition of objects, operator overloading, friend function, inheritance and polymorphism, file I/O, exception handling, templates etc.
    ii. Be able to build C++ classes using appropriate encapsulation and design principles.

- Improve your problem solving skills.
- Ultimate goal: to make you a good programmer.

Learning Outcomes:

Upon the completion of Object Oriented Programming with C++ Laboratory, the student will be able to:

- Apply an object oriented approach to programming and identify potential benefits of object-oriented programming over other approaches.
- Design applications which are easier to debug, maintain and extend.
- Apply object-oriented concepts in real world applications.

Course Content Delivery Mechanism

- Amid COVID-19, the lab will be operated in online mode till 12th August 2022, hence students are advised tomake necessary arrangement for uninterrupted internet connectivity.
- List of assignments will be shared in the google classroom. Then a brief discussion on how to solve the assignment will be done by the faculty member.
- The program execution will be verified (towards end) or corrected /debugged (any point of time) by the technical assistants/faculty via screen sharing using video communication app(Team Viewer).
- For debugging, individual meeting id may be created for the technical assistants and the faculty member and the ids will be shared through the whatsapp group. In round robin manner 25 students can be assigned to one teacher(faculty/ technical assistant)
- Lab attendance will be taken using video communication app during the lab hours (at any point of time), so students are advised to be online throughout the session.
- To ensure class participation, at any point of time , may be after 30 mins or more students may be asked to write the solution to anyone of the assignment using pen and paper and submit it in the goggle classroom within 15 to 20 minutes .Failed to do so will mark the student absent and no further evaluation will be permitted for the current lab.
- G++ compiler will be used for compiling and running the C++ programs.

Evaluation Mechanism

Sr#	Area	Mark	#	Total
1	**Internal Sending**			
1.1	Continuous Evaluation	2	15	30
1.2	Quiz	5	2	10
1.3	Viva/Activity	1	10	10
1.4	Program Test	2	5	10
Total				60
2	**End Term**			
2.1	Program Test	10	2	20
2.2	Viva	10	1	10
2.3	Quiz	10	1	10
Total				40

List of Programs/Assignments

Lab 1. Topic: Structures

i. WAP to input name, roll number and marks in 5 subjects for a student, and display it.

ii. WAP to input name, roll number and marks in 5 subjects for n number of students. Write functions to:-
   a. Find total marks and percentage of all n students.
   b. Display details of a student with a given roll number.
   c. Display the details for all the students having percentage in a given range.
   d. Sort the array in ascending order of marks.

iii. WAP to enter id, name, age and basic salary of n number of employees. Calculate the gross salary of all the employees and display it along with all other details in a tabular form, using pointer to structure.

[Gross salary= Basic salary + DA + HRA,
DA = 80% of Basic salary
HRA=10% of Basic salary]

Lab-2

Topic: Simple C++ programs using Classes and Objects

i. WAP to display the message "hello" followed by your name on screen.

ii. Create a class which stores name, roll number and total marks for a student. Input the data for a student and display it.

iii. Modify the program ii) to store marks in 5 subjects. Calculate the total marks and percentage of a student and display it.

iv. Create a class complex which stores real and imaginary part of a complex number. Input 10 complex numbers and display them.

v. Create a class distance which stores a distance in feet and inches. Input 2 distance values in objects, add them, store the resultant distance in and object and display it.
[Write the above program in two ways.
a) store the resultant distance in the calling object:C3.add(C1,C2)
b) return the resultant object C3=C1.add(C2)

vi. Create a class which stores id, name, age and basic salary of an employee. Input data for n number of employees. Calculate the gross salary of all the employees and display it along with all other details in a tabular form.
[Gross salary= Basic salary + DA + HRA,
DA = 80% of Basic salary
HRA=10% of Basic salary]

vii. Create a class which stores x and y coordinates of a point. Calculate distance between two given points and display it.

Lab-3
Topic: Friend functions & Static members

i. WAP to swap private data member of two classes.
   [The classes have no relation with each other].
ii. Create two classes which stores distance in feet, inches and meter, centimeter format respectively. Write a function which compares distance in object of these classes and displays the larger one.
iii. Create a class with an integer data member. Include functions for input and output in class. Count the number of times each function is called and display it.
iv. Create a class which stores name, roll number and total marks for a student. Input data for n students. Find the average marks scored by n students, store it as a data member of the class and display it using a function which may be called without object.
v. Create a class which stores name, author and price of a book. Store information for n number of books. Display information of all the books in a given price range using friend function.

Lab-4

Topic: Function overloading, default arguments, inline functions, reference variables

i. WAP to find area of a circle, a rectangle and a triangle, using concept of function overloading.
ii. WAP to find volume of a sphere, a cylinder and a cuboid, using function overloading.
iii. WAP which displays a given character, n number of times, using a function. When the n value is not provided, it should print the given character 80 times. When both the character and n value is not provided, it should print '*' character 80 times.
[Write the above program in two ways:-
-using function overloading.
-using default arguments.]
iv. WAP to find square and cube of a number using inline function.
v. WAP to swap two integers using pass by reference.
vi. WAP to increment the value of an argument given to function.

Lab-5

Topic: Constructors and Destructors

i. Create a class complex which stores real and imaginary part of a complex number. Include all types of constructors and destructor. The destructor should display a message about the destructor being invoked. Create objects using different constructors and display them.
ii. Create a class which stores time in hh:mm format. Include all the constructors. The parameterized constructor should initialize the minute value to zero, if it is not provided.
iii. Create a class which stores a sting and its length as data members. Include all the constructors. Include a member function to join two strings and display the concatenated string.
iv. WAP to demonstrate the order of call of constructors and destructors for a class.
v. WAP to count number of objects created from a class using concept of static data members and static member function.

Lab-6

Topic: Inheritance

i. WAP to demonstrate all types of inheritance.

ii. Create a class student which stores name, roll number and age of a student. Derive a class test from student class, which stores marks in 5 subjects. Input and display the details of a student.

iii. Extend the program ii. to derive a class from result from classs 'test' which includes member function to calculate total marks and percentage of a student. Input the data for a student and display its total marks and percentage.

iv. Extend the program ii. to include a class sports, which stores the marks in sports activity. Derive the result class from the classes 'test' and 'sports'. Calculate the total marks and percentage of a student.

v. Create a class 'shape'. Derive three classes from it: Circle, Triangle and Rectangle. Include the relevant data members and functions in all the classes. Find the area of each shape and display it.

vi. Create a class which stores employee name,id and salary Derive two classes from 'Employee' class: 'Regular' and 'Part-Time'. The 'Regular' class stores DA, HRA and basic salary. The 'Part-Time' class stores the number of hours and pay per hour. Calculate the salary of a regular employee and a par-time employee.

vii. Create a class which stores account number, customer name and balance. Derive two classes from 'Account' class: 'Savings' and 'Current'. The 'Savings' class stores minimum balance. The 'Current' class stores the over-due amount. Include member functions in the appropriate class for

-deposit money
-withdraw [For saving account minimum balance should be checked.]
[For current account overdue amount should be calculated.]
-display balance

Lab-7

Topic: Constructors in Inheritance

i. WAP to demonstrate the order of call of constructors and destructors in case of multiple inheritance.
ii. WAP to demonstrate the order of call of constructors and destructors in case of multi-level inheritance.
iii. WAP to demonstrate the order of call of constructors and destructors in case of virtual base class .
iv. Extend the program ii. of inheritance to include a class sports, which stores the marks in sports activity. Derive the result class from the classes ‘test’ and ‘sports’. Create objects using parameterized constructors .Calculate the total marks and percentage of a student.
v. Rewrite the assignment vii. From Inheritance including the parameterized constructors in all the classes.

Lab-8

Topic: Operator Overloading

i. WAP to overload following operators for class distance, which stores the distance in feet and inches.
   a) Binary + to
      -add two objects (D3=D1+D2)
      -Add an object to an integer, where the integer should be added to the inches value (D2=4+D1)
   b) Unary -
ii. Create a class to store an integer array. Overload insertion and extraction operator to input and display the array elements.
iii. Create a class which a complex number. Add two objects and display the resultant object. Overload the ++ (post and pre) operator for the class.
iv. Create a class which allocates the memory for a string through dynamic constructor. Overload the binary + to concatenate two strings and display it.
v. WAP to add two objects of time class. Overload the operator '==' to compare two objects and display whether they are equal or not. Overload the assignment operator.
vi. WAP to add two objects of distance class. Overload the operator '>' to compare two objects and return the object with larger time value and display it. Overload the '==' operator to compare and display whether two given objects contain same distance value.

Lab-9

Topic: Dynamic Binding (Virtual Functions)

i. Create a class shape. Derive three classes from it; Circle, Square and Triangle. Find area of each shape and display it, using virtual function.

ii. Create a class which stores employee name,id and salary Derive two classes from 'Employee' class: 'Regular' and 'Part-Time'. The 'Regular' class stores DA, HRA and basic salary. The 'Part-Time' class stores the number of hours and pay per hour. Calculate the salary of a regular employee and a par-time employee, using virtual function

iii. Create a class which stores account number, customer name and balance. Derive two classes from 'Account' class: 'Savings' and 'Current'. The 'Savings' class stores minimum balance. The 'Current' class stores the over-due amount. Include member functions in the appropriate class for

-deposit money
-withdraw [For saving account minimum balance should be checked.]
[For current account overdue amount should be calculated.]
-display balance

Display data from each class using virtual function.

iv. WAP to demonstrate use of pure virtual function and abstract base class.

Lab-10

Topic: File Handling

i. WAP to display content of a file using character output finction.
ii. WAP to copy content of one file to another.
iii. WAP to write 10 strings into a file and display them from file.
iv. WAP to display content of a file in reverse order.
v. WAP to count following in a given file:
    a. No. of characters
    b. No. of words
    c. No. of lines
    d. No. of uppercase letters, lowercase letters, digits and special symbols.
vi. WAP to convert all uppercase letters in a given file, to lower case and vice-versa.
vii. WAP to read and write objects to a file, using read and write functions.
viii. WAP to write multiple data to a file using insertion operator and read data from file using extraction operator.

Lab-11

Topic: Templates,Exception Handling

i. WAP to find sort an integer array and a float array, using function template.
ii. WAP to display data of two different types using function template with multiple arguments.
iii. Rewrite program i. using class template
iv. Rewrite program ii. using class template
v. WAP to throw and handle ‘division by zero’ exception.
vi. WAP to throw and handle’ array out of bound’ exception.
vii. WAP to demonstrate multiple catch and catch all.