LAB MANUAL

Object Oriented Programming (OOP) Lab.

(Code-2095) in C++

FOR U.G B.TECH THIRD SEMESTER STUDENTS

Prepared By

Prof. Naliniprava Behera

Asst.Professor, School of Computer Engineering

School of Computer Engineering

KIIT Deemed to be University, Bhubaneswar

INTRODUCTION

- Programming cannot be learned by watching others do it. Students must spend numerous hours working on programs themselves.
- This laboratory manual is a tool that will allow students to experiment with computer science & this is the beginning. As students progress through each laboratory, they may wonder how or why something works. The best way to discover the answer is to try things out.
- The purpose of this lab. manual is to acquaint the students to know the programming language as well as developing programming skills using C language.

STRUCTURE OF THIS LAB. MANUAL

- This lab. manual provides study aids from programming assignments to scheduled exercises using prepared materials.
- This lab manual is divided into **10 laboratory classes**. The laboratory classes consist of the following components:
  a) **Lab. Excerise (LE):** These are the **assignments** that ask each student to independently create small programs **during the lab time**.
  b) **Home Exercise (HE):** These are the assignments to be done during lab time by each student if lab. assignments are completed before lab. time or may be assigned as **post-lab homework** and submitted in the next lab class
  c) **Round Exercise (RE):** These are the **group assignments/small projects** to be done by each group round the time and to be submitted one copy per group at any time if asked to submit after one week of RE given or at the end of the course completion.

The approach of the Lab proceedings: **LE-HE-RE**

INSTRUCTIONS FOR STUDENTS

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

1. **General instructions**
   - Once you 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. HE3.5 (3.5 means 5th assignment of 3rd lab), then name the program as HE35_progname.c or HE35_prog.c etc.
2. **Attendance:** Attendance is required at all labs without exception. There are no make-up labs in this course. Performance will be judged based on the experiments conducted, quality and punctual submission of the labs reports for each experiment. Faculty/Instructor will take attendance. Failure to be present for an experiment will result in loosing entire marks for the corresponding lab. However, genuine cases may be considered for repeat lab. If a student misses a lab session due to unavoidable circumstances can provide a legitimate proof as soon as possible, he/she may be then be allowed by the lab instructor, to make-it-up.
3. **Laboratory Report:** At the end of every lab student will be assigned to write-up one of the experiment's problem. Your report must present a clear and accurate account, results you obtained. Student should develop habit to submit the laboratory report/assignments continuously and progressively on the scheduled dates and should get the assessment done.
4. Read the write up of each experiment to be performed, a day in advance. Understand the purpose of experiment and its practical implications.
5. Student should not hesitate to ask any difficulty faced during conduct of practical / exercise.
6. The student shall study all the questions given in the laboratory manual and practice to write the answers to these questions.
7. Student shall develop the habit of evolving more ideas, innovations, skills etc. those included in the scope of the manual.
8. Student should develop the habit of not to depend totally on teachers but to develop self learning techniques.
9. While entering into the LAB students should wear their ID cards.
10. Shut down your system after you have finished with your experiment.

PROCEDURE FOR EVALUATION

The entire lab course consists of 100 marks. The marking scheme is as follows

Continuous Evaluation marks	60
End Sem. Lab Examination	40
Total	**100**

Scheme for continuous evaluation

Students will be evaluated bi-weekly. Minimum 6 evaluations should be conducted for each student. Each evaluation carries 10 marks. The scheme is as follows:

Program & Execution	5
Lab. Record	3
Viva-Voce	2
Total	**10**

Scheme for end sem lab examination

End sem. lab exam will be conducted after the completion of all the weekly exercises. The student will not be allowed for exam if he/she is found short of attendance and has not completed all the experiments. The marking scheme for end sem lab exam is as follows:

Write-up of program	15
Program execution & Checking Results for all inputs	15
Final Viva-Voce	10
Total	**40**

CONTENTS

Lab-1: Basic Input / Output in C++

- The **objective** is that the students will learn to use the cin object to take input from the user, and the cout object to display output to the user, similar to scanf and printf in programming language C.
- **C++ Input:** In C++, cin takes formatted input from standard input devices such as the keyboard, with the help of >> operator.
- **C++ Output:** In C++, **cout** sends formatted output to standard output devices, such as the screen, with the help of with the << operator.
- **Outline of C++ Program**

```
#include <iostream>
using namespace std;
int main()
{
    .......
    .......      C++ statements/ code
    .......

    return 0;
}
```

  - ❖ The inclusion of **iostream header file** that allows to display input/output.
  - ❖ The cout object is defined inside the std namespace. To use the std namespace, **using namespace std;** statement is used. For first lab, the students need to remember this, they will learn abou why this ststement is used in later lab.classes.
  - ❖ If we don't include the using namespace std; statement, we need to use std::cout instead of cout and std::cin instead of cin.
- **Execution of C++ Program in g++ Compiler**

Step-1: Create a file named as **first.cpp** in **gedit** or any editor and write the following program code in it, then save the file and quit from gedit window.

Step-2: Compile the C++ program file named as **first.cpp**

$ g++ first.cpp

It compiles the file first.cpp, if it is error free, then go for execution to get output, else open the file again in gedit to correct the errors, again compile it till it does not show any errors.

Step-3: To get the output do the following

$ ./a.out

PROGRAM CODE

```
//Program to to add two numbers input through theboard.
#include <iostream>
using namespace std;
int main()
{
   int a, b, sum;
   cout<<"Welcome to C++ Programming Laboratory\n";
   cout<<"\nEnter two numbers:";
```

```
    cin>>a>>b;
    sum=a+b;
    cout<<"\nThe sum of the given two numbers is "<<sum;
    return 0;
}
```

INPUT/OUTPUT

RUN-1

```
Welcome to C++ Programming Laboratory

Enter two numbers: 3 5
The sum of the given two numbers is 8
```

RUN-2

```
Welcome to C++ Programming Laboratory

Enter two numbers: 10 6
The sum of the given two numbers is 16
```

Want Output in the following format	Corresponding cout statement
3+5=8 10+6=16	cout<<a<<"+"<<b<<"="<<sum;
The sum of 3 and 5 is 8	cout<<"\nThe sum of "<<a<<" and "<<b<<" is "<<sum;

- The **endl** manipulator is used to insert a new line.
  cout<<"Welcome to C++ Programming Laboratory\n"; This statement can also be written with endl as follows:
  cout<<"Welcome to C++ Programming Laboratory"<<endl;
- **Comments in C++:**
  - **Single-line comments (informally, C++ style),** start with // and continue until the end of the line. If the last character in a comment line is a \ the comment will continue in the next line.
  - **Multi-line comments (informally, C style),** start with /* and end with */ .

- **Execution of C++ Program in online Compiler:** Refer the following links for online compilation. But it is advisable to use g++ compiler to run your programs.

1. https://www.onlinegdb.com/online_c++_compiler
2. https://www.tutorialspoint.com/compile_cpp_online.php
3. https://www.codechef.com/ide
4. http://cpp.sh/
5. https://www.programiz.com/cpp-programming/online-compiler/

Lab-1: Basic Input / Output in C++ Program Excerise

Lab. Exercises (LE)

LE1.1 Write a program in to find size of fundamental data types in C++ labguage.

LE1.2 Write a program to check if two given non-negative integers have the same last digit.

LE1.3 Write a program to find out all roots of a quadratic equation.

LE1.4 Write a program in to check whether a number is prime or not.

LE1.5 Write a program to find the Greatest Common Divisor (GCD) of two numbers.

LE1.6 Write a program to find the sum of the series 1 + 1/2^2 + 1/3^3 + ..+ 1/n^n. n is the user input.

LE1.7 Write a program to find the smallest and largest elements in a given array of n integers.

LE1.8 Write a program to find out the prime factors of a number entered through keyboard by using a user define function for this.

/*Hints: A prime number is any number with no divisors other than itself and 1, such as 2 and 5. Any number can be written as a product of prime numbers in a unique way (except for the order). These are called prime factors of a number. In otherwords, In number theory, the prime factors of a positive integer are the prime numbers that divide that integer exactly, without leaving a remainder. The process of finding these numbers is called integer factorization, or prime factorization. • Enter a number : 100 • The prime factors of 100 are 2(2) and 5(2) • That is, 100 = 2 x 2 x 5 x 5, and those numbers are primes.*/

LE1.9 Write a program to store the author, title, price of n books & display the book details. n is the user input.

Home Exercise (HE)

HE1.1 Write a program in to check the upper and lower limits of integer.

HE1.2 Write a program to check two given integers, and return true if one of them is 10 or if their sum is 10.

HE1.3 Write a program to create a new string with the last char added at the front of a given string of length 1 or more.

HE1.4 Write a program to display all the prime numbers within a given range by writing a suitable user defined function for testing of the prime number.

HE1.5 Write a program to find the last prime number occur before the entered number.

HE1.6 Write a program to find the sum of the series 1 +11 + 111 + 1111 + .. n terms.

HE1.7 Write a program to find the most occurring element in an array of integers.

HE1.8 Write a program for processing students records using structure. The o/p must come in the following format with decreasing order of their average marks.

ROLL	NAME	AVERAGE	RANK
001	Harapriya	95.46	1
029	Sampad	92.34	2

HE1.9 Define a structure to store the employee name, id, date of birth, basic salary of an employee. Write a program to store n employee details. Display the employee information having highest salary. Sort the employee details according to the basic salary.

Lab-2: Input / Output in C++ with Some Basic Concepts

(Call by Value, Reference & Address, Function Overloading, Function with default arguments, Global & Local Variables, Scope Resolution Operator)

Program Excerise

Lab. Exercises (LE)

LE1.1 Write a program to demonstrate the concept of call-by-value, call-by-reference & call-by-address by taking swapping of two numbers as an example.

LE1.2 Write a program to demonstrate the use of scope resolution operator(::) by declaring same varibale name globally and locally and display the value of gloabl and local variables.

LE1.3 Write a program to find out the area of a circle and volume of a sphere by using function overloading concept. (Use one function name as AREA-VOL)

LE1.4 Write a program to find out area or volume of an shape/object by using one function name as FUN-AREA only. If the function takes one argumnet then it will calculate area of a circle, two argument for area of rectangle and three argumnet volume of a box. If no argumnet is given to the function, then it will display appropriate message.

LE1.5 Write a program to find the summation of three numbers by using one function only with function name SUM having three arguments. If at run time one argumnet is given to the function SUM, then second and third argument will be assumed by default as 10 and 20 respectively. If two argumnets are given at run time, then thrird argument will be assumed as 20. Use function with default argument concepts.

Home Exercises (HE)

HE2.1 Write a rogram to compute absolute value of a number (integer or float) by using function overloading for computation of absolute value. The number is entered through keboard.

HE2.2 By changing the number of arguments or by having different types of arguments to a function, function overloading can be achieved. Write a program to demonstrate these two concepts by taking addition of numbers as an example. Use ADD function name for function overloading purpose.

HE2.3 Modify program no.LE1.5 that if no argument will be given it will display error message and the program will work for both integer and float data types.

Lab-3: Class & Objects

Lab. Exercises (LE)

LE3.1 Define a class called Triangle, with two fields as base and height to calculate the area of a triangle. The class contains two methods such as getData() to initialize the fields of a triangle and areaTrin() to calculate the area of a triangle. Write a program to find out area of 10 triangles.

LE3.2 Write a program to enter the code, name and price of items. The user must feed the quantity in which he wants a product. The program must calculate and display the final bill of 10 nitems. In the following format.

Sl. No.	Code	Name	Price	Quantity	Total
1.	rib001	Printercatrige	300	2	600
2.	Pap45	A4 size paper	200	0	0
3.	Bk2109	Computer book	350	5	1750

Total = Rs.2350/-

LE3.3 Write a program that adds two complex numbers. The objects must be passed as function arguments.

LE3.4 Create a class TIME with members hour, minute and second. Write a program that adds two time objects.The objects must be passed as function arguments.

LE3.5 Define a class named as FRACTION that contains two data members that represent the numerator and denominator of a fraction. By defining necessary member functions, write a program to add, subtract and multiply two fractions. The add accepts the objects using call-by- value technique, subtract using call-by-reference and multiply using call-by-address technique.

Sample input/Output

For Fraction-1
Enter the numerator: 3
Enter the denominator: 5

For Fraction-2
Enter the numerator: 4
Enter the denominator: 9

Result of addition = 47/45
Result of subtraction = 7/45
Result of multiplication = 4/15

Lab-3: more on Class & Objects

Home Exercises (HE)

HE3.1 Define a class to represent a bank account. Include the following members:

Data Members

a) Name of the depositor
b) Account number
c) Type of account
d) Balance amount in the account

Member Functions

a) To assign initial value
b) To deposit an amount
c) To withdraw an amount after checking the balance
d) To display name and balance

Write a main program to test the program.

HE3.2 Create a 'DISTANCE' class with the following members:

Data Members

a) feet and inches

Member Functions

a) To input distance
b) To output distance
c) To add two distance objects

Write a main function to create objects of DISTANCE class. Input two distances and output the sum.

HE3.3 Write a program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number.

a) ADD (a , s2) - where a is an integer (real part) and s2 is a complex number.
b) ADD (s1, s2) - where s1 and s2 are complex numbers.

Lab-4: FRIEND FUNCTION & FRIEND CLASS

Lab. Exercises (LE)

LE4.1 Write a program to find out the area of an rectangle through friend function and member function separately of Rectangle class.

LE4.2 Write a program to find out the area of one rectangle and one square by using a suitable friend function named as area() for it.

LE4.3 Write a program to find out the greatest between two numbers defined in two different classes by using friend function.

LE4.4 Create a class 'COMPLEX' to hold a complex number. Write a friend function to add two complex numbers. Write a main function to add two COMPLEX objects.

LE4.5 Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of Dm with another object of DB. Use friend function to carry out the operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should in the format of feet and inches or meters and centimeters depending on the object on display.

Home Exercises (HE)

HE4.1 Write a program to find the sum of two numbers declared in a class and display the numbers and sum using friend function.

HE4.2 Write a program to find out greatest among three numbers. Define two classes, first class KBC contain two data members a and b, and second class BBC only one data member c. Input data to the data members by defining appropriate member functions. A member function max() defined in second class BBC to find out the greatest among three numbers. Solve this problem by using friend function.

HE4.3 Write a program to find out the area of an rectangle be defining a class named as Rectangle having length and breadth as its data members. Declare another class named as RectArea having one member function named as area to calculate the area by taking the data length and breadth from the Rectangle class. Solve this by using friend class.

HE4.4 Write a program to exchange values between two classes by using friend function.

HE4.5 Write a program by using friend function to add data objects of two different classes ABC and XYZ.

Lab-5: CONSTRUCTORS

Lab. Exercises (LE)

LE5.1 Write a program to perform addition of two complex numbers using constructor. Complex numbers are given through user input.

LE5.2 Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imaginary parts to equal values and third which takes two argument is used to initialize real and imaginary to two different values.

LE5.3 Create a 'DISTANCE' class with the following members:

Data Members

b) feet and inches

Member Functions

d) To input distance through constructor
e) To output distance
f) To add two distance objects

Write a main function to create objects of DISTANCE class. Input two distances (one through implicit call and other explicit call and output the sum.

LE5.4 Create a class called 'TIME' that has three integer data members for hours, minutes and seconds, a constructor to initialize the object to some constant value, member function to add two TIME objects, member function to display time in HH:MM:SS format. Write a main function to create two TIME objects, add them and display the result in HH:MM:SS format.

LE5.5 Write a program using copy constructor to copy data of an object to another object.

Home Exercises (HE)

HE1.1 Write a program to generate a fibonacci series using copy constructor.

HE1.2 Write a program to create a class for counting the number of objects created and destroyed within various block using static data member, constructor and destructors.

HE1.3 A book shop maintains the inventory of books that are being sold at the workshop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If itis, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of their quested copies is displayed otherwise the message “Required copies not in stock” is displayed. WAP using a class called Books with suitable member functions and constructors.

Lab-6: INHERITANCE

Lab. Exercises (LE)

LE6.1 (Single Inheritance) Write a program to create a class CIRCLE with one field as radius, used to calculate the area of a Circle. Create another class RECTANGLE used to calculate the area of the rectangle which is a subclass of CIRCLE class. Use the concept of single inheritance such that the radius of circle class can be re-used as length in rectangle class. Take necessary data members and member functions for both the classes to solve this problem. All the data members are initialized through the constructors. Show the result by accessing the area method of both the classes through the objects of rectangle class.

LE6.2 (Multilevel Inheritance) Modify the program no. LE6.1 as follows:
Derive a class named as BOX from RECTANGLE class. Take necessary data & member functions for this box class to calculate the volume of the box. All the data members are initialized through the constructors. Show the result by accessing the area method of circle and rectangle and the volume method of box class through the objects of box class.

LE6.3 (Hierarchical Inheritance) Modify the program no. LE6.1 as follows:
Derive a class named as CYLINDER from CIRCLE class. Take necessary data & member functions for this class to calculate the volume of the cylinder. Show the result by accessing the area method of circle and rectangle through object of rectangle class and the area of circle and volume method of cylinder class through the objects of cylinder class.

LE6.4 (Multipath Inheritance) Design a base class called Student with two fields:- (i) Name (ii) roll number. Derive two classes called Sports and Exam from the Student base class. Class Sports has a field called s_grade and class Exam has a field called e_grade which are integer fields. Derive a class called Results which inherit from Sports and Exam. This class has a character 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 usage of these classes in main.

Home Exercises (HE)

HE1.1 Write a program to create a class called STUDENT with data members Roll Number, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields a semester, fees and stipend. Enter the data for at least 5 students. Find the average age for all UG and PG students separately.

HE1.2 Create a base class called Shape with two data members of double type which could be used to compute the area of figures. Derive two specific classes called Triangle and Rectangle from the base class Shape. Add to the base class, a member function called GetData to initialize base class data members and another member function displayArea to compute and display the area of figures. Make displayArea a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes design a program which will accept dimensions of a triangle or rectangle interactively and display the area.

Lab-7: Operator Overloading

Lab. Exercises (LE)

LE7.1 Write a program to overload unary (-) operator.

LE7.2 Write a program to overload unary (-) operator using friend function

LE7.3 Write a program to overload unary (++) operator and unary (--) operator.

LE7.4 Write a program to overload unary (++) operator and unary (--) operator using friend function.

LE7.5 Write a program to design a class representing complex numbers and having functionality of performing addition and multiplication of two complex numbers using operator overloading.

LE7.6 Re-write the program no.7.5 using friend function.

LE7.7 Implement a class String containing the following functions:

- Overload + operator to carry out concatenation of strings.
- Overload = operator to carry out string copy.
- Overload =+ operator to carry out the comparison of strings.
- Function to display the length of a string

LE7.8 Write a program to create a class having two integer data members, now initialize and then add 10 to each data member using operator function where operator function can be called when you write the following statement inside main function a1+5, 10+a;

LE7.9 Write a program to create a class player having data members as player number, name, score. Now enter details of player by overloading extraction operator (>>) and display all the details by overloading insertion operator (<<).

LE7.10 Write a program to overload new and delete operator.

Home Exercises (HE)

HE1.1 Write a C++ program to create a class called MATRIX using a two- dimensional array of integers. Implement the following operations by overloading the operator== which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the operators + and -. Display the result by overloading the operator <<. if(m1==m2) Then m3=m1-m2 and m4=m1+m2 else display error.

HE1.2 Create a 'MATRIX' class of size m X n. Overload the '+' operator to add two MATRIX objects. Write a main function to implement it.

HE1.3 Derive a class 'MAT' from MATRIX class created in the previous program. Add a member function to overload '*' operator to multiply two objects. (Single Inheritance).

HE1.4 Write a C++ program to create a class called DATE. Accept two valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators + and -. After every operation display the results by overloading the operator <<.

a) no_of_days=d1-d2; where d1 and d2 are DATE objects d1>=d2 and no_of_days is an integer.
b) d2=d1-no_of_days; where d1 is a DATE object and no_of_days is an integer.

HE1.5 Write a program to create a class called OCTAL which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.

I . OCTAL h=x; where x is an integer.

II. int y=h+k; where h is an octal object and k is an integer.

Display the OCTAL results by overloading the operator <<. Also Display the values of h and y.

HE1.6 Program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

a) STRING s1= "KIIT"
b) STRING s2= "KISS
c) STRING s3= s1+s2; (use copy operator).

HE1.7 Write C++ program to create a class called STACK using array of integers. Implement the following operations by overloading the operators + and --.Also display the status and contents of the stack after each operation, by overloading the operator <<. i) s1=s1+element; where s1 is a object of the class STACK and element is an integer to be pushed on top of the stack ii) s1=s1--;where s1 is a object of the class STACK, -- operator pops the element. Handle the STACK empty and STACK full conditions.

Lab-8: POINTERS, VIRTUAL FUNCTION AND POLYMORPHISM

Lab. Exercises (LE)

LE8.1 (Pointers to objects) Define a class Item that is used to store and display the information regarding the item number and its price. Write a program to store and display the details of one items by using both normal object and pointer to object separately. Display appropriate message wherever necessary.

LE8.2 Modify the program LE8.1 by creating an array of n objects using pointers. Show the details of n items by using pointers to object concept.

LE8.3 (Pointers to derived classes) Write a program to illustrate how pointers to a base class is used for both base and derived class.

LE8.4 (this Pointer) Write a program to display the contains of an object, when local variable's name is same as member's name by using this pointer.

LE8.5 (Virtual Function) Define a class ABC. Derive two classes BBC and KBC from ABC. All the classes contains same member function name as display(). The base class pointer always holds the derived class objects.

a) Write a program such that base class pointer or reference will always access/call the base version of the members available in derived class, do not have any access to the derived class members.

b) Write a program such that base class pointer or reference will always access/call the derived version of the members available in derived class, do not have any access to the base class members.

Write down the concepts used for bit a) and b) separately.

LE8.6 (Pure Virtual Function) Write a program by modifying LE8.5 b) by making display() as pure virtual function.

Lab-9: TEMPLATES

Lab. Exercises (LE)

LE1.1 (Function Template) Define a function template for finding the minimum value contained in an array. Write main() function to find the minimum value of integer array and minimum value of floating point numbers in an array.

LE1.2 (Function Template) Write a program to define the function template for swapping the two items of various data types such as integers, float and characters etc.

LE1.3 (Function Template) Write a template function to search for a given key element from an array. Illustrate how you perform search in integer, character as well as double arrays using the same template function.

LE1.4 (Non-type as function parameters) Write a program to demonstrate the concept behind function templates with non-type as function parameters by taking sorting an array of numbers as an examples.

LE1.5 (Non-type as template parameters) Write a program to demonstrate the concept behind function templates with non-type as template parameters by taking sorting an array of numbers as an examples.

LE1.6 (Overloading Function Template) Write a C++ program illustrating overloading function template by taking sum of numbers (max. Three) entered through keyboard.

LE1.7 (Class Template) Write a program to define a class template for reading two data items from the keyboard and find out their sum.

LE1.8 (Class Template) Write a class template to represent a generic vector. Include member functions to perform the following tasks:

- ✓ To create the vector.
- ✓ To modify the value of a given element.
- ✓ To multiply the vector by a scalar value.
- ✓ To display the vector in the form (10, 20, 30,…..)

LE1.9 (Class Template) Write a program to explain class template by creating a template T for a class named pair having two data members of type T which are inputted by a constructor and a member function get-max() return the greatest of two numbers to main. Note: the value of T depends upon the data type specified during object creation.

LE1.10 (Class Template) Design a generic stack class which can be used to create integer, character or floating point stack objects. Provide all necessary data members and member functions (push, pop, display & default constructor) to operate on the stack.

LE1.11 (Class Template) Design a template Stack which can work with either a Student record or an Employee record. A Student record contains name, rollNo and cgpa. An Employee record contains name, empId and salary fields. Provide push, pop, display functions to the template stack class.]

LE1.12 (Class Template) Program to create a class called QUEUE with the member functions to add and to delete an element from the queue. Using these functions, implement a queue of integer and double. Demonstrate the operations by displaying the contents of the queue after every operation.

Lab-10: FILE HANDLING IN C++

LE10. 1 Write a program that takes student data (roll number, name, gender, marks in three subjects (phy, chem. & math) from the keyboard and store it to a file student.txt.

LE10. 2 Write a program to open the file student.txt and display the students information on the computer screen along with the total marks scored by the student.

LE10. 3 Write a program that opens two files in sequence for writing the data. Then it opens two files in sequence for reading the data and display the data in computer screen. File1 contains state names of india and File2 contains the capital of different states of India.

LE10. 4 Write a program to count number of characters, tabs, newline characters available in agiven text file.

LE10. 5 Write a program to count number of words and lines available in a file student.txt.

LE10. 6 Write a program to copy the content of two files and merge them into a third file.

N.B: Error (If any) if found, please inform over mail to npbehersfcs@kiit.ac.in or nalini.behera@gmail.com / over phone : +91-9938853966 (WhatsApp/Call).