Lab No.6 Multiple and Multilevel Inheritance

6.1 Introduction

This lab covers multiple and multilevel inheritance. Also, multi-file programming is covered.

6.2 Objectives of the lab

1. Understand the concept of multiple and multilevel inheritance.
2. Write two level inherited classes.
3. Write a class inherited from multiple base classes.
4. Write multi-file programs covering inheritance.

6.3 Activities

Perform these activities in C++, Java, and Python.

6.3.1 Activity [Multilevel Inheritance] [write in all three languages]

Create a class **First.** It contains one protected data member *f* and one public input function *f_input()*. Use the function to take *f* from user on runtime.

Next, create a derived class **Second** from **First** class. This class also contains only one protected data member *s* and one public input function *s_input*(). Call *f_input*() function inside *s_input*() and then take *s* from user on runtime.

Finally, create another derived class **Third** from **Second** class. This class contains one protected data member *t*. It contains three public functions. An input function *t_input()* that takes *t* from user on runtime, a max function *max*() that finds maximum of *f*, *s*, and *t* and displays the maximum, and show function that displays *f*, *s*, and *t*. Note, call *s_input*() inside t_input() and then take *t* from user.

Write main function to test the functionality. Create an object of **Third.** Call t_input(), show(), and max() functions according to test case given in 6.4.

Note: For python, keep same name for input function i.e. *in1*() in all three classes.

6.3.2 Activity [Multiple Inheritance] [write in C++ and Python]

Create a class **base.** It contains one protected data member *ba* and two public functions *input_base()* and *show_base*(). Use *input_base*() to take *ba* from user on runtime while *show_base*() to display content of *ba*.

Create another class **exponent.** It also contains one protected data member *exp* and two public functions *input_exp()* and *show_exp*(). Use *input_exp*() to take *exp* from user on runtime while *show_exp*() to display content of *exp*.

Next, create derived class **power** from **base** class and **exponent** class. This class contains one data member *po*. It contains three public functions. A constructor to initialize po with 1, an input function *in1()*, and *show1*() function. The *in1*() calls *input_base*() and *input_exp*() functions. The *show1*() calls *show_base*() and *show_exp*() functions; computes power using *ba* and *exp* and store in *po*; and displays computed power.

Write main function to test the functionality. Create an object of **power.** Call in1() and show1()functions according to test case given in 6.4.

Note: Write code for C++ and Python for this activity. Java does not support multiple inheritance.

6.3.3 Activity [Multi-file Programming] [write in C++ and Python]

Redo Activity 6.3.1 and 6.3.2 using multi-file programming.

Note: In C++, create header file (*.h) for each class and main file for main function (e.g. lab6t3.cpp). Include the header file to access the respective class.

Note: In python, save all class in separate *.py file (e.g. lab6t1.py containing First, Second, and Third classes and lab6t2.py containing base, exponent, and power classes)and then access using import in main python file (e.g. lab6t3.py).

Note: Since, Java is already doing multi-file programming so no need to do this activity in Java.

6.4 Testing

Test Cases for Activity 6.3.1

Sample Inputs	Sample Outputs
Declare Third object t1. Call t_input() function to give following values. Call show() function to display given values. Call max() function. Test for numbers of your choice and show results.	 Enter Number 1: 6 Enter Number 2: 4 Enter Number 3: 8 First Number is 6 Second Number is 4 Third Number is 8 8 is the maximum.

Test Cases for Activity 6.3.2

Sample Inputs	Sample Outputs
Declare power object p1. Call in1() function to give following values.	 Enter Base: 2 Enter Exponent: 3

Call show1() function to display given values. Test for base and exponent of your choice and show results.	Base: 2 Exponent: 3 Power: 8

6.5 References

1. **Class notes**
2. **Object-Oriented Programming in C++ by *Robert Lafore***
3. **How to Program C++ by *Deitel & Deitel***
4. **Programming and Problem Solving with Java by *Nell Dale & Chip Weems***
5. **Murach's Python Programming by *Micheal Urban & Joel Murach***