OBject oriented programming with java

Lab manual

Aymen khan

Aymkhan498@gmail.com

**Course Title:** Object Oriented Programming

**Course Code:**

**Course Outcomes:**

At the end of the course the student should be able to:

1. Apply Object Oriented Programming concepts to solve a given problem.
2. Apply design patterns to design a solution for a given problem.
3. Apply inheritance, polymorphism and exception handling mechanism to implement reusable, robust java programs.
4. Implement user interface java programs for a given scenario.

**List of Practical**

|  |  |  |
| --- | --- | --- |
| **Sr.#** | **Week** | **Topics** |
| 1 | One | Introduction to IDE-NetBeans, Getting Started with Java |
| 2 | Two | Java Basics (Input/output, variable declaration and initialization, strings and arrays) |
| 3 | Three | Java Basics (Selection Structure and Iterative Structure, functions) |
| 4 | Four | OOP (creating classes, objects, constructors) |
| 5 | Five | access modifiers, inheritance |
| 6 | Six | multiple/multilevel inheritance |
| 7 | Seven | Function overriding |
| 8 | Eight | Polymorphism |
| 9 | Nine | abstract classes and interfaces |
| 10 | Ten | exception handling and Java file handling |

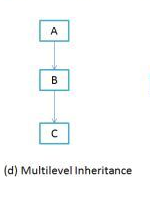
**Introduction to NetBeans and Java**

**LAB-1**

**OOP (multilevel inheritance)**

**LAB-6**

Multilevel inheritance refers to a mechanism in OO technology where one can inherit from a derived class, thereby making this derived class the base class for the new class. As you can see in below flow diagram C is subclass or child class of B and B is a child class of A.



**Example 1:**

Class X

{

public void methodX()

{

System.out.println("Class X method");

}

}

Class Y extends X

{

public void methodY()

{

System.out.println("class Y method");

}

}

Class Z extends Y

{

public void methodZ()

{

System.out.println("class Z method");

}

public static void main(String args[])

{

Z obj = new Z();

obj.methodX(); //calling grand parent class method

obj.methodY(); //calling parent class method

obj.methodZ(); //calling local method

}

}

**Example 2:**

class Shape {

public void display() {

System.out.println("Inside display");

}

}

class Rectangle extends Shape {

public void area() {

System.out.println("Inside area");

}

}

class Cube extends Rectangle {

public void volume() {

System.out.println("Inside volume");

}

}

public class Tester {

public static void main(String[] arguments) {

Cube cube = new Cube();

cube.display();

cube.area();

cube.volume();

}

}

**Task-1**

Write Java program to create a programmer derived from employee which is himself derived from person using Multilevel Inheritance. GetData(), GetEmpData(), GetProgData() functions take user inputs for Person, Employee and Programmer respectively. Display(), DisplayEmp() and DisplayProg() functions print data on scree for Person, Employee and Programmer respectively.



**Task-2**

In a bank, different customers have savings account. Some customers may have taken a loan from the bank. So, bank always maintains information about bank depositors and borrowers.

Design a Base class Customer (name, phone-number). Derive a class Depositor (accno, balance) from Customer. Again, derive a class Borrower (loan-no, loan-amt) from Depositor. Write necessary member functions to read and display the details of ‘n’ customers.

**Assignment Question**

Write a java code to implement multilevel inheritance for following classes.

