Lab 4

1. Create one superclass HillStations and three subclasses Manali, Mussoorie, Gulmarg. Subclasses extend the superclass and override its location() and famousFor() method. i.call the location() and famousFor() method by the Parent class’, i.e. Hillstations class. As it refers to the base class object and the base class method overrides the superclass method; the base class method is invoked at runtime. ii.call the location() and famousFor() method by the all subclass’,and print accordingly. --

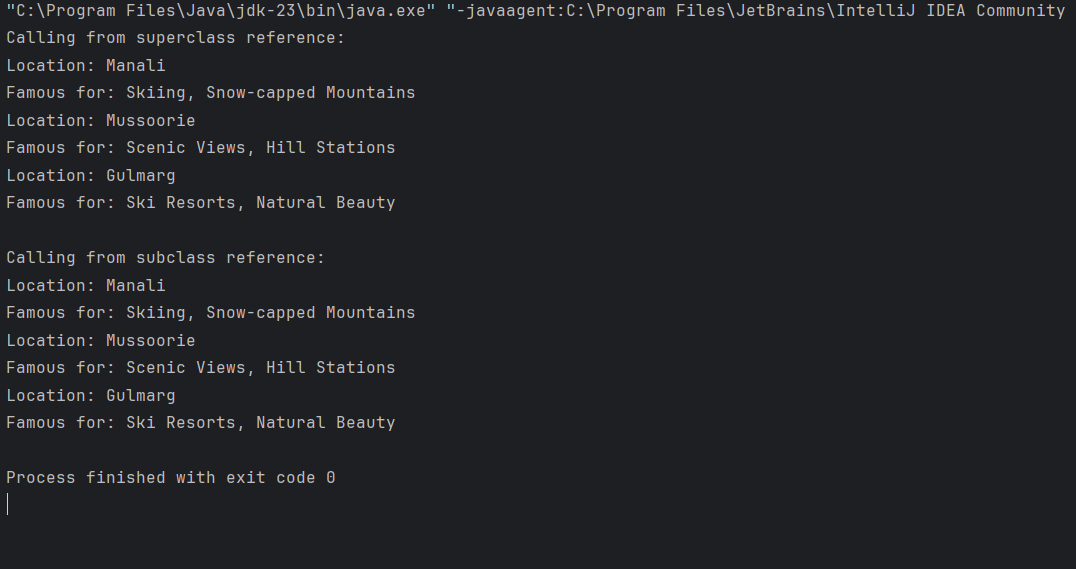
public class HillStations { // HillStations class  
 public void location() {  
 System.*out*.println("Location: Unknown");  
 }  
  
 public void famousFor() {  
 System.*out*.println("Famous for: Unknown");  
 }  
}

public class Manali extends HillStations { //Manali class  
 @Override  
 public void location() {  
 System.*out*.println("Location: Manali");  
 }  
  
 @Override  
 public void famousFor() {  
 System.*out*.println("Famous for: Skiing, Snow-capped Mountains");  
 }  
}

class Mussoorie extends HillStations { //Mussoorie class  
 @Override  
 public void location() {  
 System.*out*.println("Location: Mussoorie");  
 }  
  
 @Override  
 public void famousFor() {  
 System.*out*.println("Famous for: Scenic Views, Hill Stations");  
 }  
}

class Gulmarg extends HillStations { //Gulmarg class  
 @Override  
 public void location() {  
 System.*out*.println("Location: Gulmarg");  
 }  
  
 @Override  
 public void famousFor() {  
 System.*out*.println("Famous for: Ski Resorts, Natural Beauty");  
 }  
}

public class Places {  
 public static void main(String[] args) {  
 // Creating objects of each subclass  
 HillStations manali = new Manali();  
 HillStations mussoorie = new Mussoorie();  
 HillStations gulmarg = new Gulmarg();  
  
 // Calling methods from the superclass reference  
 System.*out*.println("Calling from superclass reference:");  
 manali.location();   
 manali.famousFor();   
  
 mussoorie.location();   
 mussoorie.famousFor();  
  
 gulmarg.location();   
 gulmarg.famousFor();   
  
 System.*out*.println("\nCalling from subclass reference:");  
 // Directly calling methods from subclass references  
 Manali realManali = new Manali();  
 Mussoorie realMussoorie = new Mussoorie();  
 Gulmarg realGulmarg = new Gulmarg();  
  
 realManali.location();   
 realManali.famousFor();   
  
 realMussoorie.location();  
 realMussoorie.famousFor();  
  
 realGulmarg.location();   
 realGulmarg.famousFor();   
 }  
}



2.Write a Java program that demonstrates method overriding by creating a superclass called Animal and two subclasses called Dog and Cat. ● The Animal class should have a method called makeSound(), which simply prints "The animal makes a sound." ● The Dog and Cat classes should override this method to print "TheCat/The dog meows/barks" respectively. ● The program should allow the user to create and display objects of each class.

class Animal {  
 public void makeSound() {  
 System.*out*.println("The animal makes a sound.");  
 }  
}

class Dog extends Animal {  
 @Override  
 public void makeSound() {  
 System.*out*.println("The dog barks.");  
 }  
}

class Cat extends Animal {  
 @Override  
 public void makeSound() {  
 System.*out*.println("The cat meows.");  
 }  
}

import java.util.\*;  
  
public class AnimalSound {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("Choose an animal to create : ");  
 System.*out*.println("1. Dog");  
 System.*out*.println("2. Cat");  
 int choice = sc.nextInt();  
  
 Animal animal;  
 if (choice==1) {  
 animal = new Dog();  
 } else if (choice==2) {  
 animal = new Cat();  
 } else {  
 System.*out*.println("Defaulting to Animal.");  
 animal = new Animal();  
 }  
  
 animal.makeSound();  
 }  
}

Output :

