Name =Allahdad Chachar

Section =D

CMS=023-22-0056

Lab 07

1. Create a class Employee that contains two instance variables name, and id. It contains one method Intro( ) that display name and id. Employee class derives Person class. Person class contains one instance variable grade. It contains one method Intro( ) that display name, id, and grade. Create one object of Person class that display Intro( ) method of Person class and Intro( ) method of Employee class.

CODE

class Employee

{

String name="Allahdad";

int id=123;

public void intro()

{

System.out.println("Name:"+name);

System.out.println("Id:"+id);

}

}

class Person extends Employee

{

String grade="HOD";

public void intro()

{

super.intro();

System.out.println("Name:"+name);

System.out.println("Id:"+id);

System.out.println("Grade:"+grade);

}

}

class Q1

{

public static void main(String args[])

{

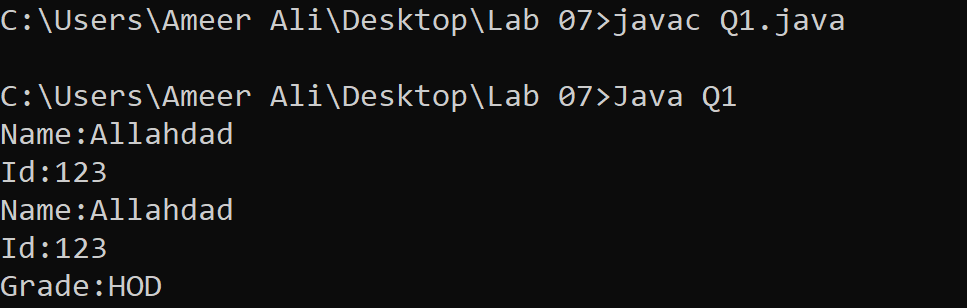
Person obj1=new Person();

obj1.intro();

}

}

OUTPUT



1. Create a class Maths that contains one instance variables radius. It contains one method display ( ) that “Hello I am display method of class Maths”. Algebra class is derived from Maths class. It contains one method display( ) that display “Hello I am display method of Algebra”. You have to perform upcasting for creating object and display the display method that show method overriding.

CODE

class Math

{

int radius;

public void display()

{

System.out.println("\nHello I am display method of class Math\n");

}

}

class Algebra extends Math

{

public void display()

{

System.out.println("Hello I am display method of class Algebra\n");

}

}

class Q2

{

public static void main(String args[])

{

Math obj1;

obj1 = new Math();

obj1.display();

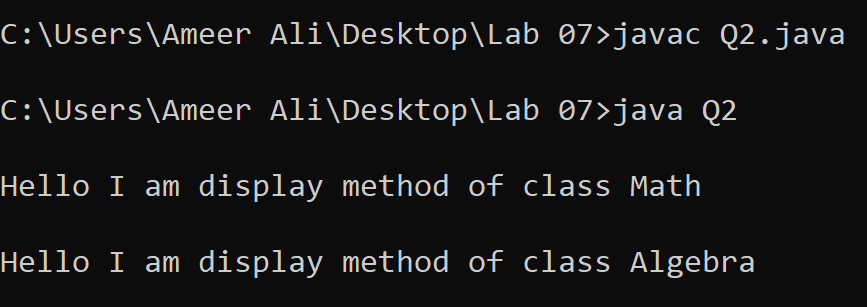
obj1 = new Algebra();

obj1.display();

}

}

OUTPUT



3.Create a parent class called "Animal" and two child classes called "Dog" and "Cat". Each child class should have a unique "makeSound" method that outputs a sound specific to the animal. Then, create an array of "Animal" objects and loop through it, calling the "makeSound" method for each animal.

CODE

abstract class Animal{

abstract public void makesound();

}

class Dog extends Animal

{

public void makesound()

{

System.out.println("Dog Sound: Bho wao");

}

}

class Cat extends Animal

{

public void makesound()

{

System.out.println("Cat Sound : Miyaon");

}

}

class Q3

{

public static void main(String args[])

{

Animal [] obj1=new Animal[2];

obj1[0]=new Dog();

obj1[1]=new Cat();

for(int i=0;i<2;i++){

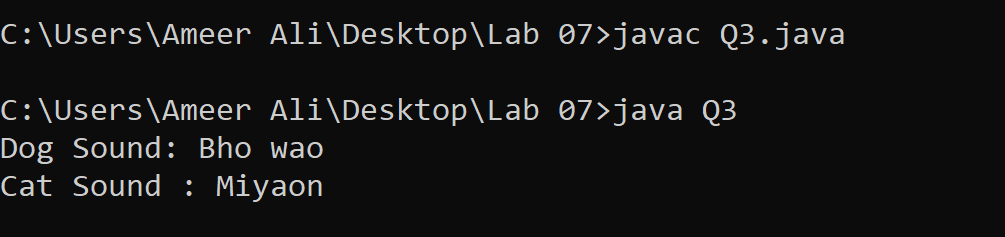
obj1[i].makesound();

}

}

}

OUTPUT



4.Create a parent class called "Shape" and three child classes called "Circle", "Rectangle", and "Triangle". Each child class should have a unique method that calculates its area. Then, create an array of "Shape" objects and loop through it, calling the "calculateArea" method for each shape.

CODE

abstract class Shape

{

abstract public void Calculate();

}

class Circle extends Shape

{

public void Calculate()

{

System.out.println("Area of circle :"+(2\*3.16\*3));

}

}

class Rectangle extends Shape

{

public void Calculate()

{

System.out.println("Area of Rectangle :"+(4\*6));

}

}

class Triangle extends Shape

{

public void Calculate()

{

System.out.println("Area of Triangle :"+(5\*6/2));

}

}

class Q4

{

public static void main(String args[])

{

Shape [] obj1=new Shape[3];

obj1[0]=new Circle();

obj1[1]=new Rectangle();

obj1[2]=new Triangle();

for(int i=0;i<3;i++)

{

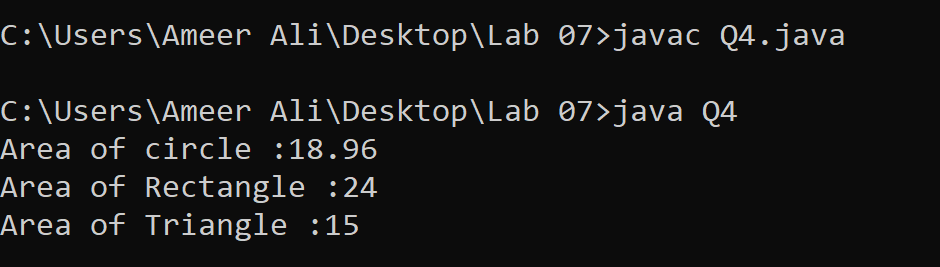
obj1[i].Calculate();

}

}

}

OUTPUT



5.Create a parent class called "Vehicle" and two child classes called "Car" and "Motorcycle". Each child class should have a unique method that prints out the number of wheels the vehicle has. Then, create an array of "Vehicle" objects and loop through it, calling the "printNumWheels" method for each vehicle.

CODE

abstract class Vehicle

{

abstract public void No\_of\_wheels();

}

class Car extends Vehicle

{

public void No\_of\_wheels()

{

System.out.println("Number of wheels of Car :"+4);

}

}

class Motorcycle extends Vehicle

{

public void No\_of\_wheels()

{

System.out.println("Number of wheels of Motorcycle :"+2);

}

}

class Q5

{

public static void main(String args[])

{

Vehicle [] obj1=new Vehicle[2];

obj1[0]=new Car();

obj1[1]=new Motorcycle();

for(int i=0;i<2;i++)

{

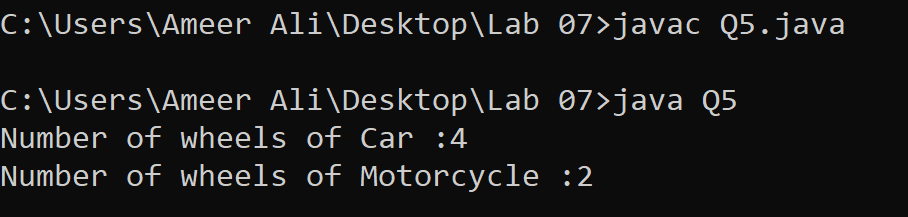
obj1[i].No\_of\_wheels();

}

}

}

OUTPUT



6.Create a parent class called "Person" and two child classes called "Teacher" and "Student". Each child class should have a unique method that calculates their salary or allowance respectively. Then, create an array of "Person" objects and loop through it, calling the "calculateIncome" method for each person.

CODE

abstract class Person

{

abstract public void Calulate\_Salary();

}

class Teacher extends Person

{

public void Calulate\_Salary()

{

System.out.println("\nTEACHER\n");

int Salary=40000;

int allowance=5000;

System.out.println("Salary:"+Salary+"\nAllowance:"+allowance+"\nTotal Salary:"+(Salary+allowance));

}

}

class Student extends Person

{

public void Calulate\_Salary()

{

System.out.println("\nSTUDENT\n");

int Salary=20000;

int allowance=2000;

System.out.println("Salary:"+Salary+"\nAllowance:"+allowance+"\nTotal Salary:"+(Salary+allowance));

}

}

class Q6

{

public static void main(String args[])

{

Person [] obj1=new Person[2];

obj1[0]=new Teacher();

obj1[1]=new Student();

for(int i=0;i<2;i++)

{

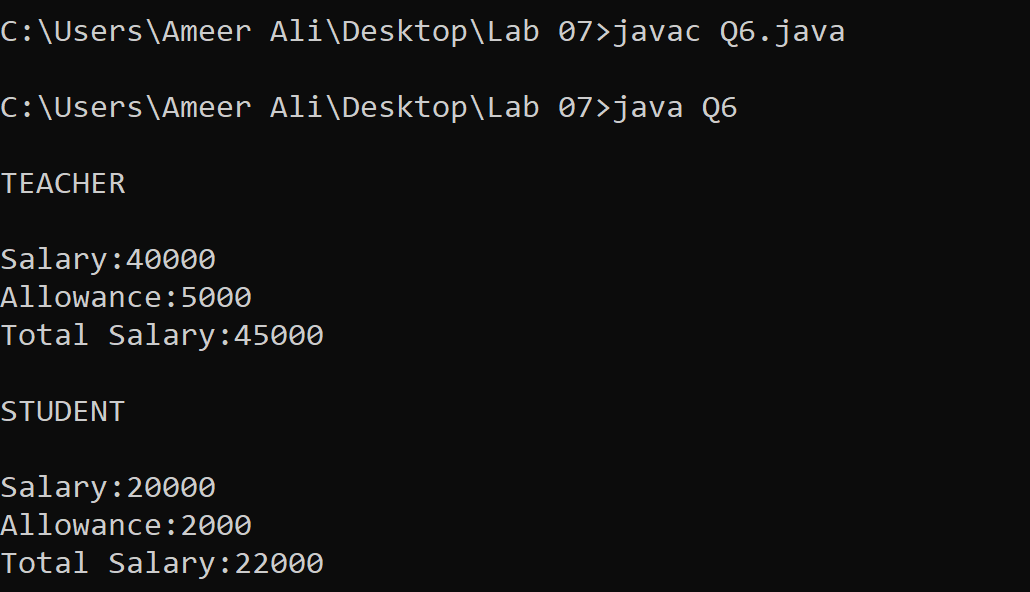
obj1[i].Calulate\_Salary();

}

}

}

OUTPUT



7.Create a parent class called "BankAccount" and two child classes called "CheckingAccount" and "SavingsAccount". Each child class should have a unique method that calculates their interest rate or fee respectively. Then, create an array of "BankAccount" objects and loop through it, calling the "calculateFees" method for each account.

CODE

abstract class Bank\_Account

{

abstract public void Intrest\_Fee();

}

class Checking\_Account extends Bank\_Account

{

public void Intrest\_Fee()

{

System.out.println("\nINTREST FEE\n");

System.out.println("Fee of checking Account is "+2000);

}

}

class Saving\_Account extends Bank\_Account

{

public void Intrest\_Fee()

{

System.out.println("Fee of Saving Account is "+1000);

}

}

class Q7

{

public static void main(String args[])

{

Bank\_Account [] obj1=new Bank\_Account[2];

obj1[0]=new Checking\_Account();

obj1[1]=new Saving\_Account();

for(int i=0;i<2;i++)

{

obj1[i].Intrest\_Fee();

}

}

}

OUTPUT

