



SUKKUR INSTITUTE OF BUSINESS ADMINISTRATION UNIVERSITY

Department of computer science

Programming Fundamentals (BSCS -II)

Spring 2024

Assignment - (08)

Submitted to Ma'am Nimra Mughal

Submitted by Mukesh Kumar

Section E"



INHERITANCE

Q1:

```
class Person{

String name;

String address;

Person(String name, String address){

this.name = name;

this.address = address;

}

String getname(){

return name;

}

String getAddress(){

return address;

}

void setAddress(String address){

this.address = address;

}

public String toString(){

return "Name = "+name+" Address = "+address;

}

}

class Student extends Person{

String program;

int year;

int fee;
```



```
Student(String name, String address, String program, int year, int fee ){  
    super(name,address);  
    this.program = program;  
    this.fee = fee;  
    this.year = year;  
}  
  
String getProgram(){  
    return program;  
}  
void setProgram(String program){  
    this.program = program;  
}  
int getYear(){  
    return year;  
}  
void setYear(int year){  
    this.year = year;  
}  
int getFee(){  
    return fee;  
}  
void setFee(int fee){  
    this.fee=fee;  
}  
@Override  
public String toString(){  
    return super.toString()+" Program = "+program+" Year = "+year+" Fee = "+fee;  
}  
}
```



```
class Staff extends Person{
    String school;
    int pay;
    Staff(String name, String address, String school, int pay){
        super(name,address);
        this.school=school;
        this.pay=pay;
    }
    String getSchool(){
        return school;
    }
    void setSchool(String school){
        this.school = school;
    }
    int getPay(){
        return pay;
    }
    void setPay(int pay){
        this.pay = pay;
    }
    @Override
    public String toString(){
        return super.toString()+" School = "+school+" Pay = "+pay;
    }
}

public class Question1{
    public static void main(String[] args){
        Person person1 = new Person("Mukesh"," Naukot");
        Student student1 = new Student("Mukesh","Naukot","BSCS",2023,100000);
        Staff teacher = new Staff("Maam Nimra","Sukkur","IBA",100000);
    }
}
```



```
System.out.println(person1);  
System.out.println(student1);  
System.out.println(teacher);  
}  
}
```

OUTPUT:

```
Name = Mukesh Address = Naukot  
Name = Mukesh Address = Naukot Program = BSCS Year = 2023 Fee = 100000  
Name = Maam Nimra Address = Sukkur School = IBA Pay = 100000  
PS C:\Users\Mukesh> |
```

Q2:

```
public class Question2 {  
    public static void main(String[] args){  
        Point2D point2d = new Point2D();  
        point2d.setX(3.2f);  
        point2d.getX();  
        point2d.setY(5.5f);  
        point2d.getX();  
        point2d.setXY(3.2f, 5.5f);  
        point2d.getXY();  
        System.out.println(point2d);  
  
        Point3D point3d = new Point3D();  
        point3d.setX(6.2f);  
        point3d.getX();  
        point3d.setY(7.5f);  
        point3d.getX();  
        point3d.setXY(6.2f, 7.5f);  
        point3d.getXY();  
    }  
}
```



```
        System.out.println(point3d);

    }

}

class Point2D{
    float x = 0.0f;
    float y = 0.0f;
    Point2D(float x, float y){
        this.x = x;
        this.y = y;
    }
    Point2D(){

    }
    float getX(){
        return this.x;
    }
    void setX(float x){
        this.x = x;
    }
    float getY(){
        return this.y;
    }
    void setY(float y){
        this.y = y;
    }
    void setXY(float x, float y){
        this.x=x;
        this.y=y;
    }
}
```



```
float[] getX(){
    float data[] = new float[2];
    data[0]=this.x;
    data[1]=this.y;
    return data;
}

public String toString(){
    return "X = "+x+" Y = "+y;
}

}

class Point3D extends Point2D{
    float z=0.0f;
    Point3D(float x, float y, float z){
        this.x=x;
        this.y=y;
        this.z=z;
    }
    Point3D(){

    }
    float getZ(){
        return this.z;
    }
    void setZ(float z){
        this.z=z;
    }
    void setXYZ(float x,float y, float z){
        setXY(x,y);
        this.z=z;
    }
}
```



```
}  
  
float[] getXYZ(){  
    float[] data = new float[3];  
    data[0]=this.x;  
    data[1]=this.y;  
    data[2]=this.z;  
    return data;  
}  
  
public String toString(){  
    return super.toString()+" Z= "+z;  
}  
  
}
```

OUTPUT:

```
X = 3.2 Y = 5.5  
X = 6.2 Y = 7.5 Z= 0.0  
PS C:\Users\Mukesh>
```

Q3:

```
public class Question3 {  
    public static void main(String[] args){  
        Point point = new Point();  
        Moveable moveable = new Moveable();  
  
        point.getX();  
        point.setX(5);  
        point.getY();  
        point.setY(7);  
        point.setXY(5, 7);  
        point.getXY();  
    }  
}
```




```
System.out.println(point);
```

```
moveable.setxspeed(10);
```

```
moveable.setyspeed(20);
```

```
moveable.getxspeed();
```

```
moveable.getyspeed();
```

```
moveable.setXyspeed(10, 20);
```

```
moveable.getXyspeed();
```

```
System.out.println(moveable);
```

```
System.out.println(moveable.move());
```

```
}
```

```
}
```

```
class Point{
```

```
private float x=1.0f;
```

```
private float y=1.0f;
```

```
Point (float x, float y){
```

```
    this.x=x;
```

```
    this.y= y;
```

```
}
```

```
Point(){
```

```
}
```

```
float getX(){
```

```
    return x;
```

```
}
```

```
void setX(float x){
```

```
    this.x=x;
```

```
}
```



```
float getY(){
    return y;
}
void setY(float y){
    this.y=y;
}
void setXY(float x, float y){
    this.x=x;
    this.y=y;
}
float[] getXY(){
    float[] data = new float[2];
    data[0] = this.x;
    data[1] = this.y;
    return data;
}
public String toString(){
    return "X = "+x+" Y = "+y;
}

}

class Moveable extends Point{
    float xspeed =1.0f;
    float yspeed =1.0f;

    Moveable(float x, float y, float xspeed , float yspeed){
        super(x,y);
        this.xspeed = xspeed;
        this.yspeed = yspeed;
    }
}
```



```
Moveable(float xspeed, float yspeed){  
    this.xspeed= xspeed;  
    this.yspeed= yspeed;  
}  
Moveable(){  
  
}  
float getxspeed(){  
    return xspeed;  
}  
void setxspeed(float xspeed){  
    this.xspeed = xspeed;  
}  
float getyspeed(){  
    return yspeed;  
}  
void setyspeed(float yspeed){  
    this.yspeed = yspeed;  
}  
  
void setXYSpeed(float xspeed, float yspeed){  
    this.xspeed = xspeed;  
    this.yspeed = yspeed;  
}  
float[] getXYSpeed(){  
    float[] data = new float[2];  
    data[0] = this.xspeed;  
    data[1] = this.yspeed;  
    return data;  
}
```



```
public String toString(){  
    return super.toString()+" Xspeed = "+xspeed+" yspeed = "+yspeed;  
  
}
```

```
Moveable move(){  
    this.setX(this.getX()+this.xspeed);  
    this.setY(this.getY()+this.yspeed);  
    return this;  
}  
  
}
```

OUTPUT:

```
X = 5.0 Y = 7.0  
X = 1.0 Y = 1.0 Xspeed = 10.0 yspeed = 20.0  
X = 11.0 Y = 21.0 Xspeed = 10.0 yspeed = 20.0  
PS C:\Users\Mukesh>
```

Q4:

```
public class Question4 {  
    public static void main(String[] args){  
        Shape shap1 = new Shape();  
        Circle circle = new Circle();  
        Rectangle rect = new Rectangle();  
        Square sq = new Square();  
        shap1.setColor("yello");  
        shap1.getColor();  
        shap1.isFilled(false);  
        shap1.isFilled();  
        System.out.println(shap1);  
    }  
}
```



```
circle.setColor("green");  
circle.getColor();  
circle.isFilled(false);  
circle.isFilled();  
System.out.println(circle);
```

```
rect.setColor("blue");  
rect.getColor();  
rect.isFilled(false);  
rect.isFilled();  
System.out.println(rect);
```

```
sq.setColor("white");  
sq.getColor();  
sq.isFilled(false);  
sq.isFilled();  
System.out.println(sq);
```

```
}  
}
```

```
class Shape{
```

```
    String color ="red";  
    boolean filled = true;  
    Shape(){  
  
    }
```



```
Shape(String color, boolean filled){
    this.color = color;
    this.filled = filled;
}

String getColor(){
    return this.color;
}

void setColor(String color){
    this.color = color;
}

boolean isFilled(){
    return this.filled;
}

void isFilled(boolean filled){
    this.filled = filled;
}

}

public String toString(){
    return "Shape: "+ "Color = "+color+" Filled = "+filled;
}
}

class Circle extends Shape{
    float radius =1.0f;
    float PI = 3.142f;
    Circle(){

    }

    Circle(float radius){
        this.radius =radius;
    }
}
```



```
}  
  
Circle(float radius, String color, boolean filled){  
    super(color,filled);  
    this.filled = filled;  
}  
  
float getRadius(){  
    return this.radius;  
}  
  
void setRaduis(float radius){  
    this.radius = radius;  
  
}  
  
float getArea(){  
    return PI*radius*radius;  
}  
  
float getPerimeter(){  
    return 2*PI*radius;  
}  
  
public String toString(){  
    return "Circle: "+ super.toString()+" radius = "+radius;  
}  
}  
  
class Rectangle extends Shape{  
    int width = 1;  
    int length=1;  
  
    Rectangle(){  
  
    }  
  
    Rectangle(int width, int length){
```



```
this.width = width;
this.length= length;
}
Rectangle(int width, int length, String color, boolean filled){
    super(color,filled);
    this.width = width;
    this.length= length;
}
int getWidth(){
    return this.width;
}
void setWidth(int width){
    this.width = width;
}
int getLength(){
    return this.length;
}
void setLength(int length ){
    this.length = length;
}

int getArea(){
    return width*length;
}
int perimeter(){
    return 2*(width+length);
}
public String toString(){
    return "Rectangle: "+super.toString()+" Length = "+this.length+" width = "+this.width;
}
```




```
}  
  
class Square extends Rectangle{  
    Square(){  
  
    }  
    Square(int side){  
        this.length = side;  
    }  
    Square(int side, String color, boolean filled){  
        this.length = side;  
        this.color = color;  
        this.filled = filled;  
  
    }  
    int getSide(){  
        return this.length;  
    }  
    void setSide(int side){  
        this.length = side;  
    }  
    void setLength(int side ){  
        this.length = side;  
    }  
    void setWidth(int side ){  
        this.width = side;  
    }  
  
    public String toString(){  
        return "Square: "+ super.toString();  
    }  
}
```



```
}
```

OUTPUT:

```
Shape: Color = yello Filled = false
Circle: Shape: Color = green Filled = false radius = 1.0
Rectangle: Shape: Color = blue Filled = false Length = 1 width = 1
Square: Rectangle: Shape: Color = white Filled = false Length = 1 width
= 1
PS C:\Users\Mukesh>
```

Q5:

```
public class Question5 {
    public static void main(String[] args){
        Animal animal = new Animal("Animal");
        Mammal mammal = new Mammal("Mammal");
        Cat cat = new Cat("Cat");
        Dog dog = new Dog("Dog");
        System.out.println(animal);
        System.out.println(mammal);
        System.out.println(cat);
        System.out.println(dog);

    }
}

class Animal{
    String name;

    Animal(String name){
        this.name = name;
    }

    public String toString(){
        return "Animal[ Name = "+name+"]";
    }
}
```



```
}
```

```
}
```

```
class Mammal extends Animal{  
    Mammal(String name ){  
        super(name);  
    }  
  
    public String toString(){  
        return "Mammal["+super.toString()+"]";  
    }  
}
```

```
class Cat extends Mammal {  
    Cat(String name){  
        super(name);  
    }  
    void greets(){  
        System.out.println("Meow");  
    }  
    public String toString(){  
        return "Cat [ "+super.toString()+"]";  
    }  
}
```

```
class Dog extends Mammal{  
    Dog(String name){  
        super(name);  
    }  
  
    void greets(){  
        System.out.println("Woof");  
    }  
}
```



```
}

void greets(Dog another){
    System.out.println("Woof");
}

public String toString(){
    return "Dog [ "+super.toString()+"]";
}
}
```

OUTPUT:

```
Animal[ Name = Animal]
Mammal[Animal[ Name = Mammal]]
Cat [ Mammal[Animal[ Name = Cat]]]
Dog [ Mammal[Animal[ Name = Dog]]]
PS C:\Users\Mukesh>
```

HACKERRANK

Q1:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class Animal{
    void walk(){
        System.out.println("I am walking");
    }
}
```



```

}

class Bird extends Animal{

    void fly(){

        System.out.println("I am flying");

    }

    void sing(){

        System.out.println("I am singing");

    }

}

public class Solution{

    public static void main(String args[]){

        Bird bird = new Bird();

        bird.walk();

        bird.fly();

        bird.sing();

    }

}

```

OUTPUT:

The screenshot shows the HackerRank interface for the 'Java Inheritance I' challenge. The problem description states: 'Using inheritance, one class can acquire the properties of others. Consider the following Animal class: class Animal { void walk() { System.out.println("I am walking"); } }'. It then asks to create a Bird class that extends Animal and has a fly method. The solution code provided is: class Bird extends Animal { void fly() { System.out.println("I am flying"); } }. The final instruction is to create a Bird object that can both fly and walk. The solution code for the main method is: public class Solution { public static void main(String[] args) { Bird bird = new Bird(); bird.walk(); bird.fly(); bird.sing(); } }. The output shows a 'Congratulations' message and a 'Test case 0' result with the expected output: 1. I am walking, 2. I am flying, 3. I am singing.

**Q2:**

```
import java.io.*;

import java.util.*;

import java.text.*;

import java.math.*;

import java.util.regex.*;

class Arithmetic{

    int add(int a, int b){

        return (a+b);

    }

}

class Adder extends Arithmetic{

}

//Write your code here

public class Solution{

    public static void main(String []args){

        // Create a new Adder object

        Adder a = new Adder();

        // Print the name of the superclass on a new line

        System.out.println("My superclass is: " + a.getClass().getSuperclass().getName());

        // Print the result of 3 calls to Adder's `add(int,int)` method as 3 space-separated integers:

        System.out.print(a.add(10,32) + " " + a.add(10,3) + " " + a.add(10,10) + "\n");

    }

}
```

OUTPUT:



Course: Object Oriented Programming | Java Inheritance II | HackerRank | Access Account - HackerRank | Google Password Manager

hackerrank.com/challenges/java-inheritance-2/problem?isFullScreen=true

HackerRank | Prepare > Java > Object Oriented Programming | Java Inheritance II | Exit Full Screen View

Write the following code in your editor below:

1. A class named Arithmetic with a method named add that takes 2 integers as parameters and returns an integer denoting their sum.

2. A class named Adder that inherits from a superclass named Arithmetic.

Your classes should not be public.

Input Format

You are not responsible for reading any input from stdin; a locked code stub will test your submission by calling the add method on an Adder object and passing it 2 integer parameters.

Output Format

You are not responsible for printing anything to stdout. Your add method must return the sum of its parameters.

Sample Output

The main method in the Solution class above should print the

You have earned 10.00 points! 76% 133/150
You are now 17 points away from the 4th star for your java badge.

Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge

Test case 0

Compiler Message
Success

Expected Output
Download

1	My superclass is: Arithmetic
2	42 13 20

Type here to search | 35°C | ENG | 4:44 pm | INTL | 16/04/2024