# **Name: Hamad Nasir**

# **Section: BSCS-6c**

# **Roll Number: 120312**

# **Lab No: 5**

Activity One:

Here the class A has a no argument constructor which prints a statement. Class B extends A but it is empty. In the main class Activity One object is made of class b and it is calling constructor in class b. Since Class B has nothing but it is extending A that’s why the constructor of A is also of B that’s why it performed the function of constructor A that is print a statement.

Code:

class A

**{**

public A()

{

System**.**out**.**println**(** "A's no-arg constructor is invoked"**);**

}

**}**

class B extends A {}

public class ActivityOne

**{**

public static void main( String[] args)

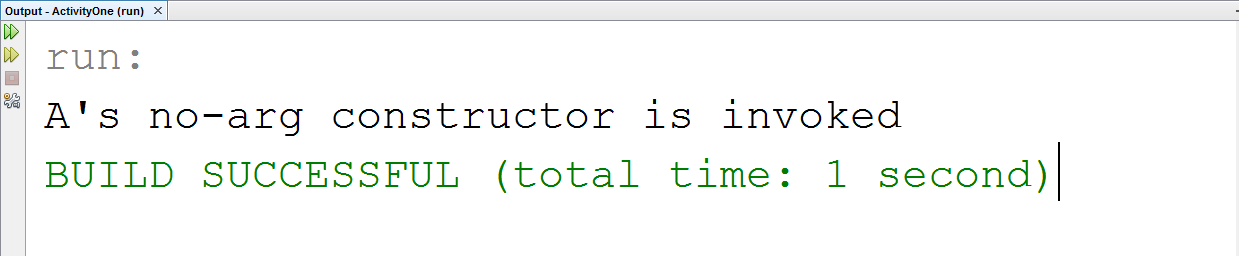
**{**

B b = new B();

}

**}**

Output:



Activity Two:

Here class A and B both have a constructor but B is extending A that means constructor of A is also extending in class B. In main class ActivityTwo object is made of class B which has argument 3 so it surely call the constructor of B which is print statement. But as class B extends class A so by default it will also call constructor of class A and will print the statement of class A.

Code:

class A

**{**

public A()

{

System.out.println( "A's constructor is invoked");

}

**}**

class B extends A

**{**

public B(int t)

{

System.out.println( "B's constructor is invoked");

}

**}**

public class ActivityTwo

**{**

public static void main( String[] args)

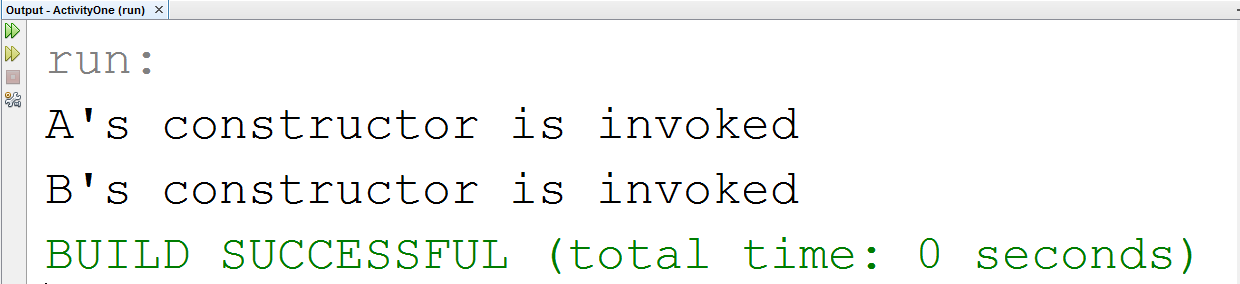
**{**

B b = new B(3);

}

**}**

Output:



Activity Three:

Corrected Code:

class A

**{**

public A( ) {}

**}**

class B extends A

**{**

public B() {}

**}**

public class ActivityThree

**{**

public static void main( String[] args)

**{**

B b = new B();

}

**}**

Concept:

The error is there because the constructor in Class A which is super class has argument which has argument but in its sub class B no argument constructor is used. And then the object of subclass B is also calling that constructor of Class A which is error. That’s why when no argument constructor is used in subclass then it important that there is also a no argument constructor in superclass.

Activity Four:

The error was stack overflow because get area in class B was itself again and again that’s why there was stack overflow error.

Corrected Code:

package activityone;

public class ActivityOne {

public static void main( String[] args) {

B b = new B( 5, 10);

System.out.println( "Area = " + b.getArea());

}

}

package activityone;

class B extends Circle

{

private double length;

B( double radius, double length)

{

super( radius);

this.length = length;

}

/\*\* Override getArea() \*/

public double getArea()

{

return super.getArea() \* length;

}

}

package activityone;

class Circle

{

private double radius;

public Circle( double radius)

{

this.radius = radius;

}

public double getRadius()

{

return radius;

}

public double getArea()

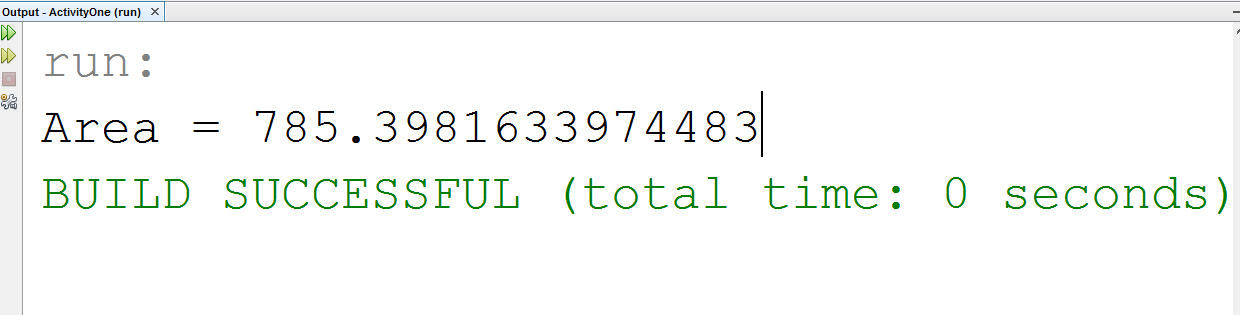
{

return radius \* radius \* Math.PI;

}

}

Output:



Task One:

Code:

public class GeometricObject {

//data fields

private String color = "white";

private boolean filled;

private java.util.Date dateCreated;

//no argument constructor

public GeometricObject(){

}

//constructor with two arguments

public GeometricObject(String color, boolean filled){

this.color = color;

this.filled = filled;

}

//get color method

public String getColor(){

return color;

}

//set color method

public void setColor(String color){

this.color = color;

}

//get filled boolean variable

public boolean isFilled(){

return filled;

}

//set filled boolean variable

public void setFilled(boolean filled){

this.filled =filled;

}

//get method for obtaining the current date

public java.util.Date getDateCreated(){

return new java.util.Date();

}

//printing

public String toString(){

return String.format("\n The color is %s \n and the filled propery is %s\n date is %s",color,filled,getDateCreated());

} //end method toString

} //end class GeometricObject

public class Rectangle extends GeometricObject {

//date field

private double width = 1.0;

private double height = 1.0;

//No argument constructor

Rectangle(){

}

//constructor whith two arguments

Rectangle (double width, double height){

this.width=width;

this.height = height;

}

// four-argument constructor

Rectangle(double width, double height, String color, boolean filled){

//invoking the constructor of super class GeometricObject

super(color,filled);

setWidth(width);

setHeight(height);

}

//get width of rectangle

public double getWidth(){

return width;

}

//set width of rectangle

public void setWidth(double width){

this.width = width;

}

//get height of rectangle

public double getHeight(){

return height;

}

//set height of rectangle

public void setHeight(double height){

this.height = height;

}

//get area of the rectangle

public double getArea(){

return height\*width;

}

//get perimeter of rectangle

public double getPerimeter(){

return 2\*(width+height);

}

//printing method toString

public String toString(){

return String.format("Rectangle: width ="+ width + " height ="+height+"\n "+super.toString());

}

} //end class Rectangle

import java.util.Scanner;

public class TestRectangle {

public static void main(String[] args){

//data fields

double width , height;

boolean filled;

Scanner input = new Scanner(System.in);

Scanner input2 = new Scanner(System.in);

//giving reference name to the object of class Rectangle

Rectangle rect;

System.out.print("Enter the Width of rectangle: ");

width = input.nextDouble();

System.out.print("Enter the height of rectangle: ");

height = input.nextDouble();

System.out.print("Enter the color of rectangle: ");

String color = input2.nextLine();

System.out.print("Is the rectangle filled or not(enter true or false): ");

filled = input2.nextBoolean();

//Creating the object of the class Rectangle

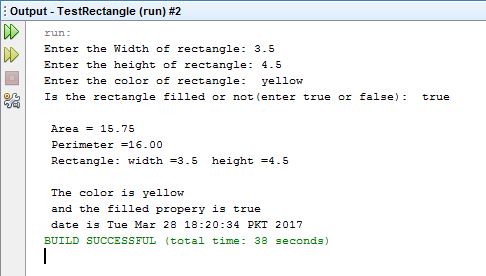
rect = new Rectangle(width,height,color,filled);

System.out.printf("\n Area = %.2f\n Perimeter =%.2f\n %s\n",rect.getArea(),rect.getPerimeter(),rect.toString());

}

}

Output:



Task Two:

Code:

package test\_task2;

public class Test\_Task2 {

public static void main(String[] args) {

Person myPerson = new Person();

System.out.println(myPerson.toString());

Student myStudent = new Student();

System.out.println(myStudent.toString());

Employee myEmployee = new Employee();

System.out.println(myEmployee.toString());

Faculty myFaculty = new Faculty();

System.out.println(myFaculty.toString());

Staff myStaff = new Staff();

System.out.println(myStaff.toString());

}

package test\_task2;

class Student extends Person {

//define class status

public static final int FRESHMEN = 1;

public static final int SOPHOMORE = 2;

public static final int JUNIOR = 3;

public static final int SENIOR = 4;

int classStatus;

//no arg constructer

Student() {

classStatus = FRESHMEN;

}

//define method toString()

public String toString() {

return "Student" + name;

}

}

package test\_task2;

class Staff extends Employee {

//define var1

String title;

//no arg constructer

Staff() {

title = "Janitor";

}

//define method toString()

public String toString() {

return "Staff" + name;

}

}

package test\_task2;

public class Person {

//define var1, var2, var3, var4

String name;

String address;

String phoneNumber;

String email;

//no arg constructer

Person() {

name = "Hamad Nasir";

address = "1234 Main St.";

phoneNumber = "(123)-456-7890";

email = "gmail@gmail.com";

}

//define method toString()

public String toString() {

return "Person" + name;

}

}

package test\_task2;

class Faculty extends Employee {

//define var1, var2

String officeHours;

String rank;

//no arg constructer

Faculty() {

officeHours = "9-10 am";

rank = "Professor";

}

//define method toString()

public String toString() {

return "Faculty" + name;

}

}

package test\_task2;

class Employee extends Person {

//define var1, var2, var 3

String office;

String salary;

//no arg constructer

Employee() {

office = "111";

salary = "$80,000";

}

//define method toString()

public String toString() {

return "Employee" + name;

}

}

package test\_task2;

class MyDate {

//define var1, var2, var 3

int year;

int month;

int day;

}

Output:

