**Name: Abdul Ghaffar Kalhoro**

**Reg: no: 194699**

**Class: BSCS-6C (2nd semester)**

**Lab-5**

**Object Oriented Programming(OOP)**

**ActivityOne:**

Here class B extends the class A through that all the methods in class A are accessible to the class B and hence in the main class ActivityOne the object of class B is created in which the no argument constructor is created as an object by default which is invoking the no argument constructor from class A.

**Source Code:**

class A {

public A()

{

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

}

}

class B extends A { }

public class ActivityOne {

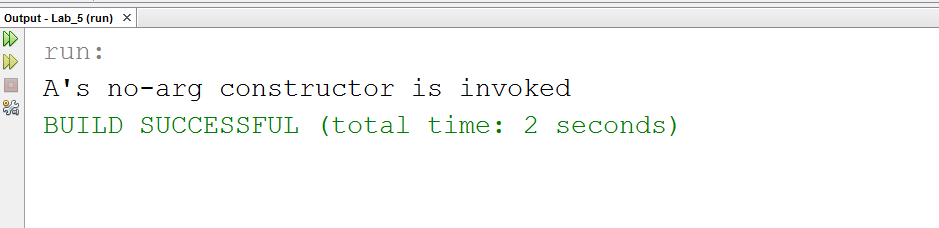
public static void main( String[] args) {

B b = new B();

}

}

**OutPut:**

****

**ActivityTwo:**

In Activity 2 the default class is called automatically and print statement used in it is printed automatically after called and then the other constructor of the class B is printed due to the creation of object in the main class ActivityTwo.

**Source 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)

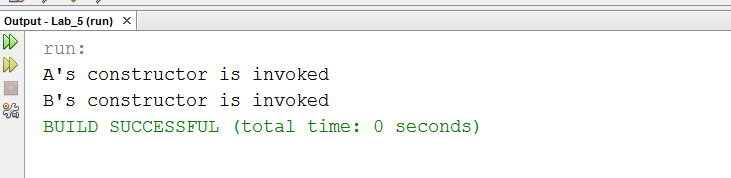
{

B b = new B(3);

}

}

**OutPut:**



**ActivityThree:**

In this java code there is no any no argument constructor in super class A but the no argument constructor used in subclass B which is not possible, we can have no argument constructor in a super class while it may not be in the sub class but the opposite is not possible.

**Corrected Source Code:**

class A

{

public A(){

}

public A( int x) {}

}

class B extends A

{

public B() {}

}

public class ActivityThree

{

public static void main( String[] args)

{

B b = new B();

}

}

**ActivityFour:**

Here in this task the error is of stack overflow in which in the method of getArea in class B calls itself recursively which causes the error.

**Corrected Source Code:**

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;

}

}

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;

}

}

//CS 212: Object Oriented Programming Page 5

public class ActivityFour

{

public static void main( String[] args)

{

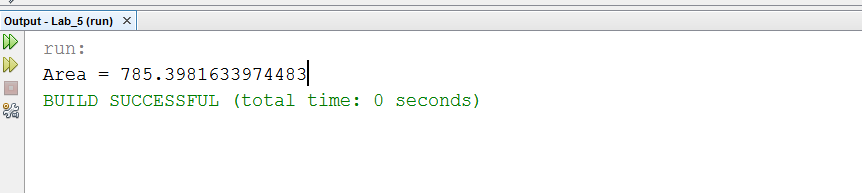
B b = new B( 5, 10);

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

}

}

**OutPut:**



**TASK\_1**

**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

**public** String getColor(){

**return** color;

} //end method getColor

//set color

**public** **void** setColor(String color){

**this**.color = color;

} //end method setColor

//get filled boolean variable

**public** **boolean** isFilled(){

**return** filled;

} //end method isFilled

//set filled boolean variable

**public** **void** setFilled(**boolean** filled){

**this**.filled =filled;

} //end method setFilled

//get method for obtaining the current date

**public** java.util.Date getDateCreated(){

**return** **new** java.util.Date();

} //end method getDateCreated

//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

//this class extends the class of 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;

} //end method getwidth

//set width of rectangle

**public** **void** setWidth(**double** width){

**this**.width = width;

} //end method setwidth

//get height of rectangle

**public** **double** getHeight(){

**return** height;

} //end method getheight

//set height of rectangle

**public** **void** setHeight(**double** height){

**this**.height = height;

} //end method setheight

//get area of the rectangle

**public** **double** getArea(){

**return** height\*width;

} //end method getArea

//get perimeter of the rectangle

**public** **double** getPerimeter(){

**return** 2\*(width+height);

} //end method getPerimeter

//printing method toString

**public** String toString(){

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

}

} //end class Rectangle

//Main class of Testing Rectangle and GeometricObject classes

**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 rect1;

System.***out***.println("Rectangle Components: ");

System.***out***.print("Enter Width: ");

width = input.nextDouble();

System.***out***.print("Enter height: ");

height = input.nextDouble();

System.***out***.print("Enter color: ");

String color = input2.nextLine();

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

filled = input2.nextBoolean();

//Creating the object of the class Rectangle

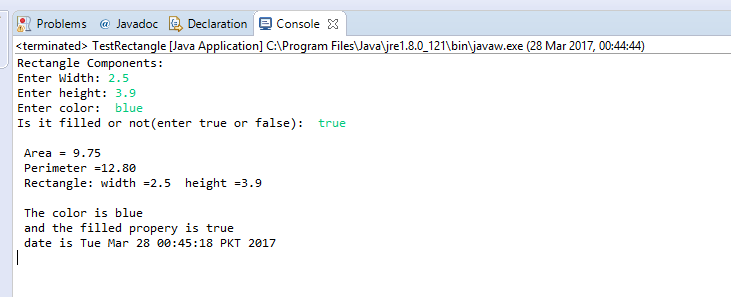
rect1 = **new** Rectangle(width,height,color,filled);

System.***out***.printf("\n Area = %.2f\n Perimeter =%.2f\n %s\n" ,rect1.getArea(),rect1.getPerimeter(),rect1.toString());

} //end main method

} //end class TestRectangle.

**OUTPUT**

****

**TASK\_2**

//Test Class for Task 2

**public** **class** TestPerson {

**public** **static** **void** main(String[] args){

Person person = **new** Person("Abdul Ghaffar", "H-12 Islamabad",

"02654698722", "abdul12@gmail.com");

Student student = **new** Student("Safdar", "12 street Karachi", "02132598752",

"safdar100@gmail.com", "FRESHMAN");

Employee employee = **new** Employee("John", "Village Abri, Kamber", "03254698799",

"john97@gmail.com", "Ideal office", 70000,"22-dec-1999");

Faculty faculty = **new** Faculty("Faraday", "98 temple street", "5987525478",

"faraday29@yahoo.com", "Dell corporation", 50000,"25-march-2010", "Professor", "4pm to 6pm");

Staff staff = **new** Staff("Einstein", "33 Lahori road", "03256987414",

"einstein65@hotmail.com","bright lenovo", 65000,

"19-april-2013","Executive Assistant");

System.***out***.println("Person:\n"+person.toString());

System.***out***.println("Student:\n"+student.toString());

System.***out***.println("Employee:\n"+employee.toString());

System.***out***.println("Faculty:\n"+faculty.toString());

System.***out***.println("Staff:\n"+staff.toString());

}

}

//class person

**class** Person {

//data fields

**private** String name;

**private** String address;

**private** String phoneNumber;

**private** String email;

//constructor

Person(String Name, String Address, String PNo, String Email){

name = Name;

address = Address;

phoneNumber = PNo;

email = Email;

}

**public** **void** setName(String N){

name = N;

}

**public** String getName(){

**return** name;

}

**public** **void** setAddress(String addr){

address = addr;

}

**public** String getAddress(){

**return** address;

}

**public** **void** setPhoneNo(String PNo){

phoneNumber = PNo;

}

**public** String getPNo(){

**return** phoneNumber;

}

**public** **void** setEmail(String Memail){

email = Memail;

}

**public** String getEmail(){

**return** email;

}

**public** String toString(){

**return** String.*format*(" Name: %s\n Address: %s\n %s: %s\n %s: %s\n",name,address,"Phone Number"

,phoneNumber,"Email",email);

}

} //end class Person.

//Student class

**class** Student **extends** Person {

**private** String status;

//constructor

Student(String Name, String Address, String PNo, String Email,String stts){

//invoking the constructor of the super class.

**super**(Name, Address, PNo, Email);

status = stts;

}

**public** **void** setStatus(String status){

**this**.status = status;

}

**public** String getStatus(){

**return** status;

}

**public** String toString(){

**return** String.*format*(" %s: %s\n %s: %s\n %s: %s\n %s: %s\n %s: %s\n","Name",getName(),"Address"

,getAddress(),"Phone Number"

,getPNo(),"Email",getEmail(),"Status",getStatus());

}

} //end class Student

//Employee class

**class** Employee **extends** Person {

**private** String office;

**private** **int** salary;

**private** String dateHired;

//constructor

Employee(String Name, String Address, String PNo, String Email,String Office,**int** sal,String date){

//invoking the constructor of the super class.

**super**(Name, Address, PNo, Email);

**this**.office = Office;

salary = sal;

dateHired = date;

}

**public** **void** setOffice(String office){

**this**.office = office;

}

**public** String getOffice(){

**return** office;

}

**public** **void** setSalary(**int** salary){

**this**.salary = salary;

}

**public** **int** getSalary(){

**return** salary;

}

**public** **void** setDateHired(String dateH){

**this**.dateHired = dateH;

}

**public** String getDateHired(){

**return** dateHired;

}

**public** String toString(){

**return** String.*format*("%s %s: %s\n %s: %d Rs.\n %s: %s\n"

,**super**.toString(),"Office",getOffice(),"Salary"

,getSalary(),"Date Hired",getDateHired());

}

} ////end class Employee.

//Faculty class

**class** Faculty **extends** Employee {

**private** String rank;

**private** String officeHours;

//constructor

Faculty(String Name, String Address, String PNo,

String Email,String Office,**int** sal,String date,String Rank,String OffHour){

//invoking the constructor of the super class.

**super**(Name, Address, PNo, Email, Office,sal, date);

rank = Rank;

officeHours = OffHour;

}

**public** **void** setRank(String rank){

**this**.rank = rank;

}

**public** String getRank(){

**return** rank;

}

**public** **void** setOfficeHours(String offH){

officeHours = offH;

}

**public** String getOfficeHours(){

**return** officeHours;

}

**public** String toString(){

**return** String.*format*("%s %s: %s\n %s: %s\n",**super**.toString(),"Rank",getRank(),"Office Hours",getOfficeHours());

}

} //end class Faculty.

//Staff class.

**class** Staff **extends** Employee {

**private** String title;

//constructor

Staff(String Name, String Address, String PNo, String Email,String Office,**int** sal,String date,String title){

//invoking the constructor of the super class.

**super**(Name, Address, PNo, Email, Office,sal, date);

**this**.title = title;

}

**public** **void** setTitle(String tit){

title = tit;

}

**public** String getTitle(){

**return** title;

}

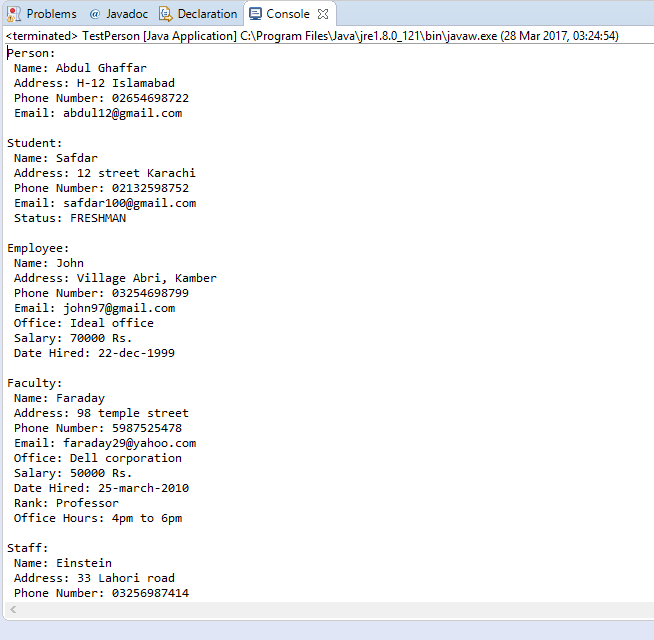
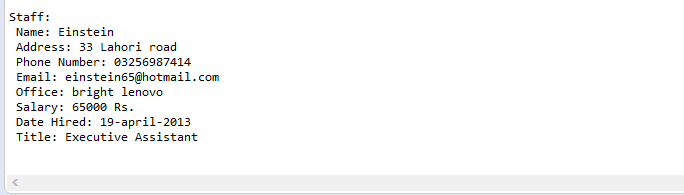
**public** String toString(){

**return** String.*format*("%s %s: %s\n",**super**.toString(),"Title",title);

}

} //end class Staff.

**OUTPUT**

****