**Muhammad Rizwan Khalid**

**180459**

**BSCS 6-A**

**Lab no: 02**

**Activity One:**

package activiy;

public class Activiy {

public static void main(String[] args) {

B b = new B();

}

}

class A{

public A(){

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

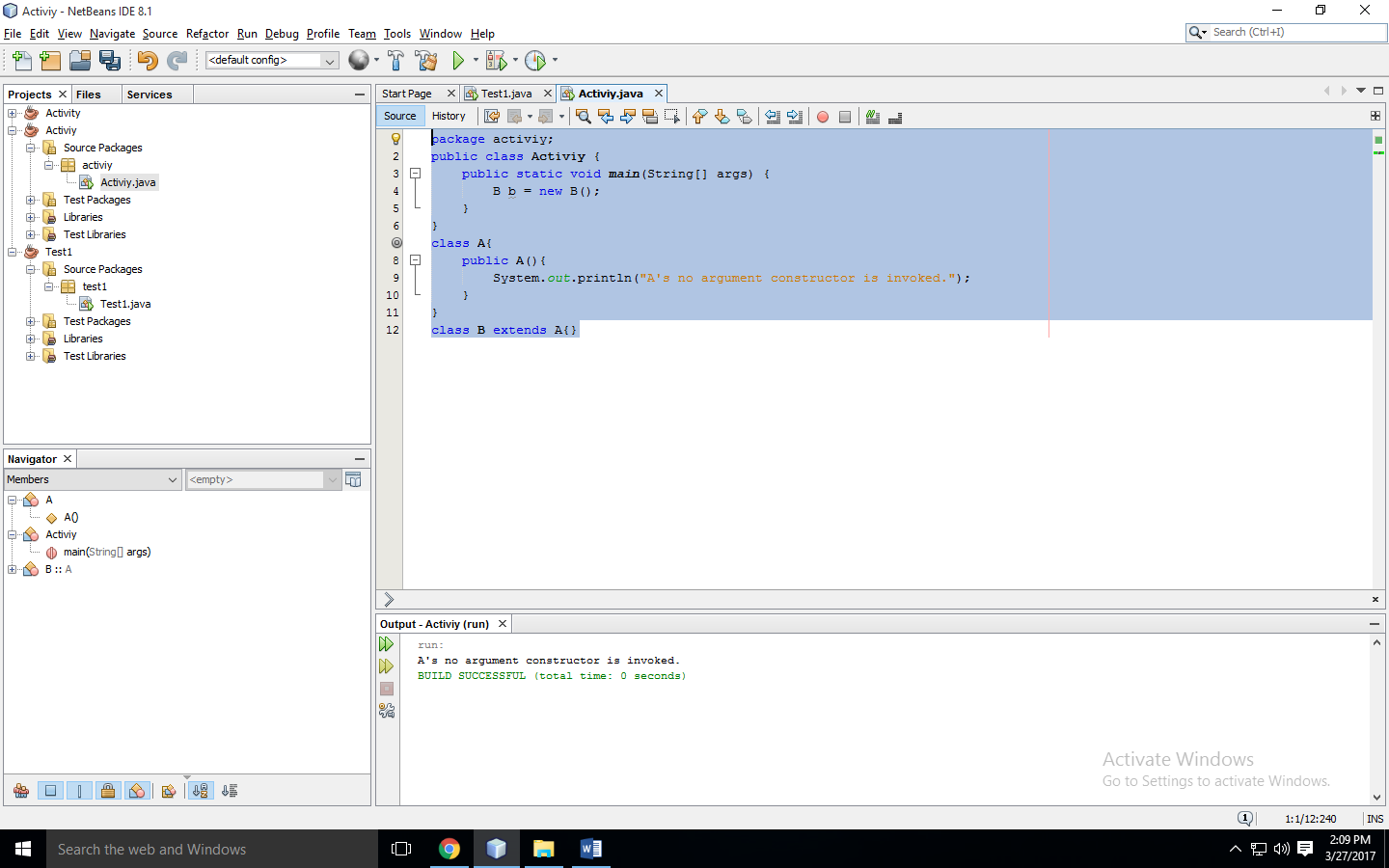
}

}

class B extends A{}

**Answer:** In the output the statement inside the constructor of class A is executed. This is the concept of inheritance. In this concept we increase the feature set of another class. In this program class B inherits the characteristics of the class A. That’s why when we created the object the output was the statement of Constructor A.

**Output:**



**Activity two:**

package activiy;

public class Activiy {

public static void main(String[] args) {

B b = new B(3);

}

}

class A{

public A(){

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

}

}

class B extends A{

public B(int t)

{

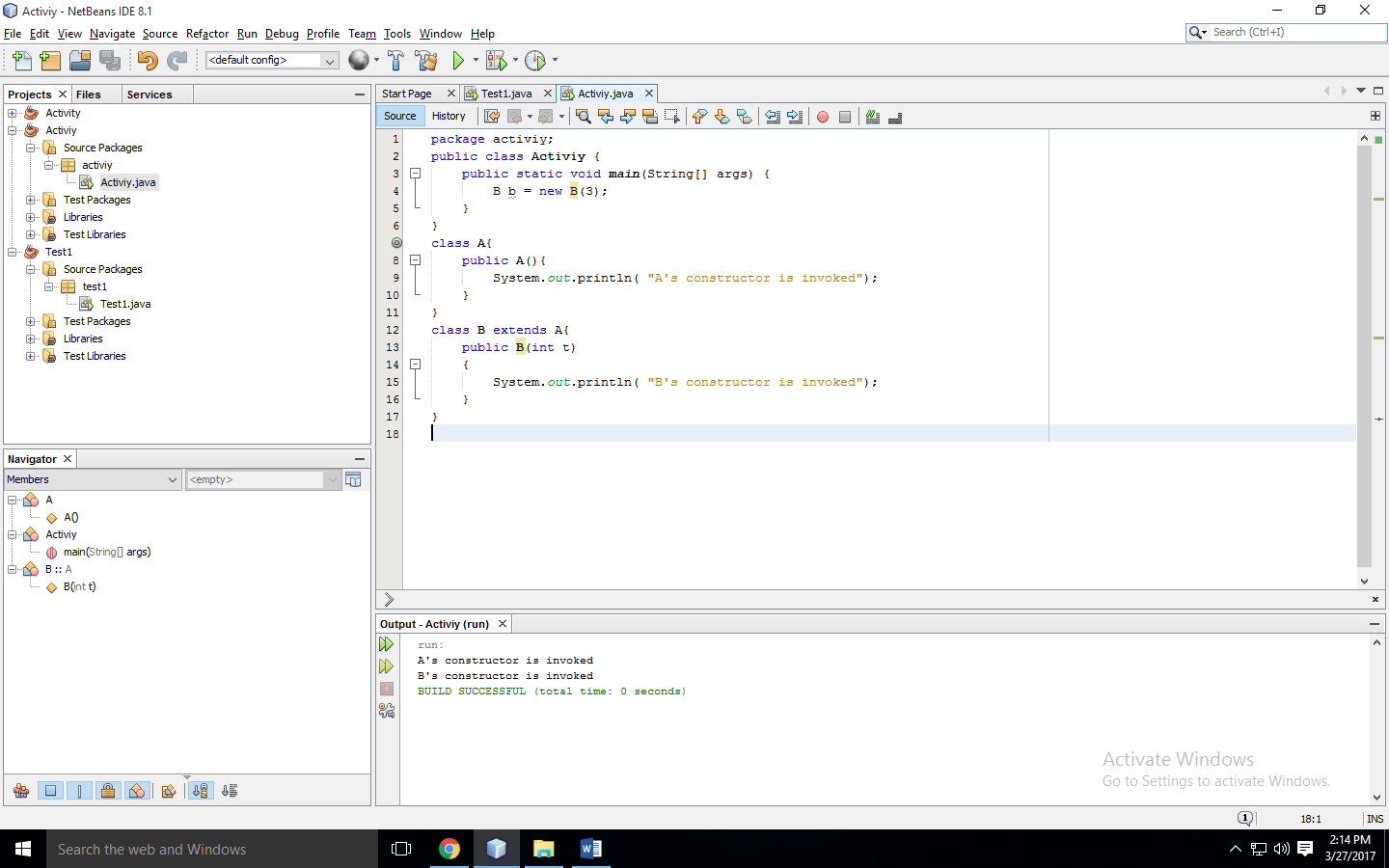
System.out.println( "B's constructor is invoked");

}

}

**Answer:** In this program we have created a super class A and the derived class B. The constructor of class B is one argument while that of A is no argument constructor. When we created the object of B, the class is extending to A so, first no argument Constructor of class A is compiled then the Constructor of B is compiled.

**Output:**



**Activity Three:**

package activiy;

public class Activiy {

public static void main(String[] args) {

B b = new B();

}

}

class A{

public A() {

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

}

class B extends A{

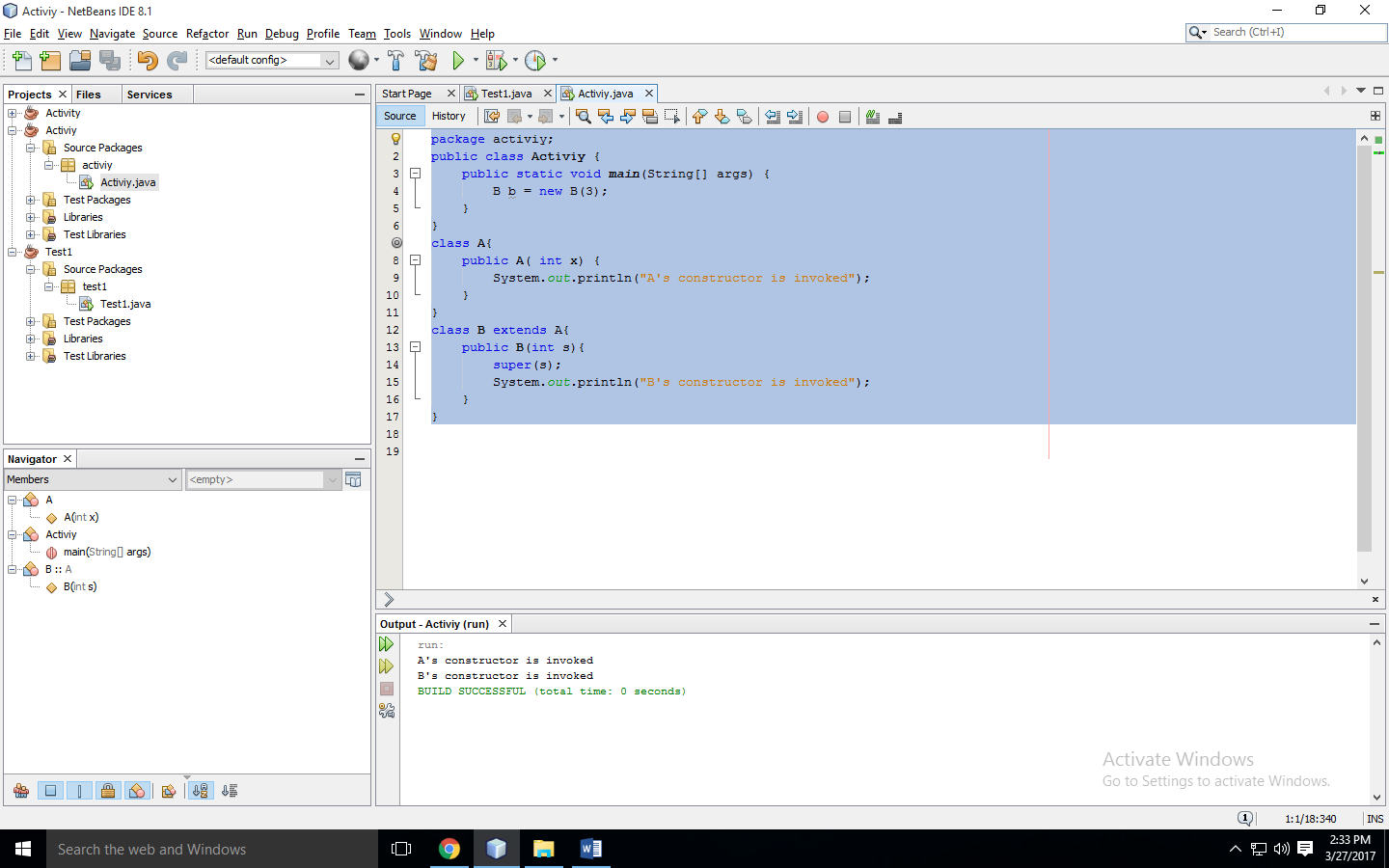
public B(){

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

}

}

**Output:**



**Output Explanation:** In Activity 3, there was an error. We were creating the object of class B by invoking no argument constructor. The class B was extending the class A which has the one argument constructor. While constructor of class B tends to call default constructor of A. That’s why we faced error. We will make the default constructor of parent class to resolve the matter.

**Activity Four:**

package activiy;

public class Activiy {

public static void main(String[] args) {

B b = new B( 5, 10);

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

}

}

class Circle{

private double radius;

public Circle(){

radius = 0;

}

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;

public B(){

length = 0;

}

public B( double radius, double length){

super(radius);

this.length = length;

}

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

@Override

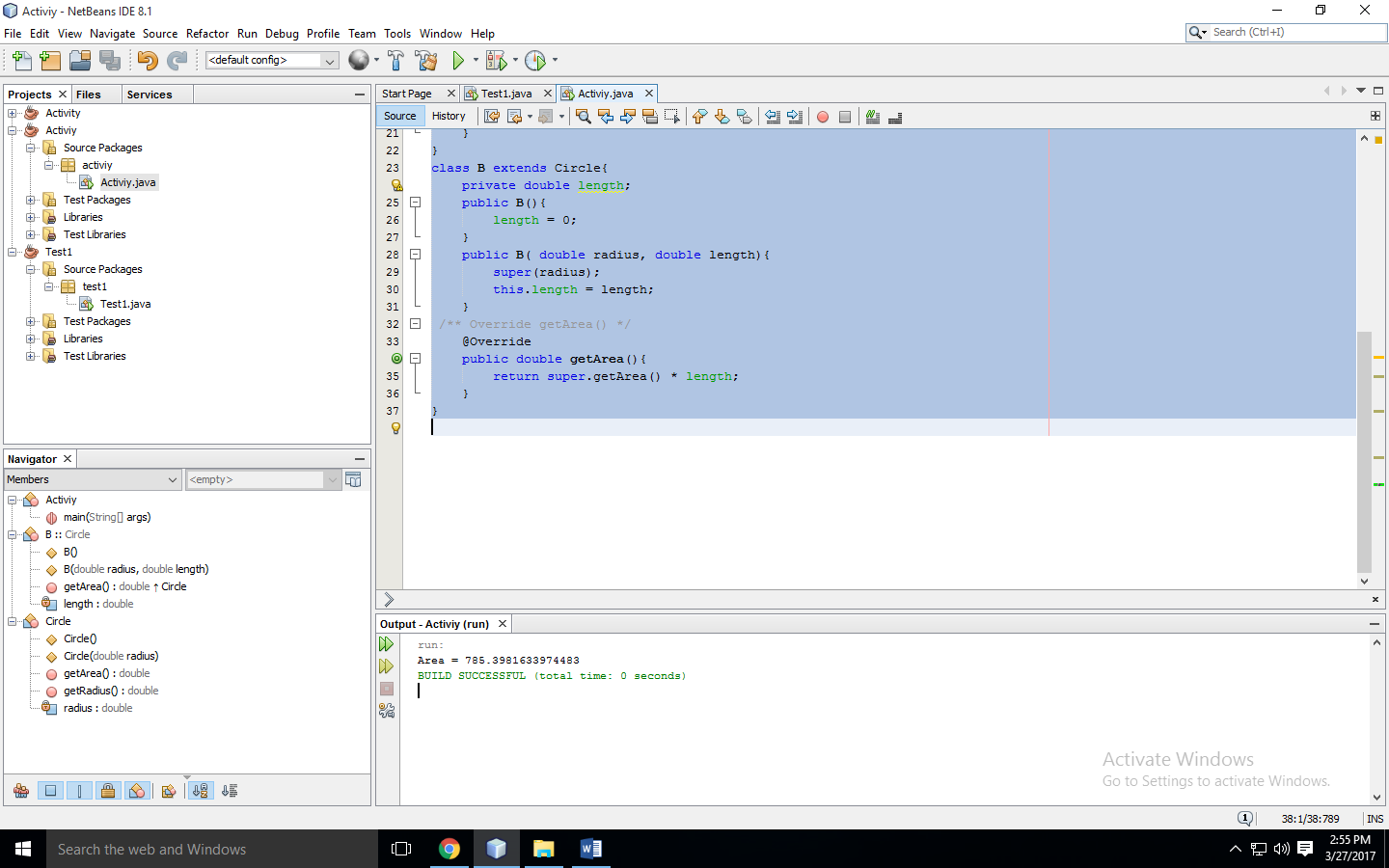
public double getArea(){

return super.getArea() \* length;

}

}

**Output:**



**Task one:**

package activiy;

import java.util.\*;

public class Activiy {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double w,h;

String s;

boolean b;

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

w = input.nextDouble();

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

h = input.nextDouble();

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

s = input.next();

System.out.print("Is the color filled? ");

b = input.nextBoolean();

Rectangle r1 = new Rectangle(w,h,s,b);

System.out.println("Area : "+r1.getArea()+"\nPerimeter: "+r1.getPerimeter()+"\nColor: "+r1.getColor()+"\nFilled: " +b);

}

}

class GeometricObject{

private String color;

private boolean filled;

private Date dateCreated = new Date();

public GeometricObject(){

color = null;

filled = false;

}

public GeometricObject(String color, boolean filled){

this.color = color;

this.filled = filled;

}

public String getColor(){

return color;

}

public void setColor(String color){

this.color = color;

}

public void setFilled(boolean filled){

this.filled = filled;

}

public String getDateCreated(){

return dateCreated.toString();

}

@Override

public String toString(){

return String.format("Color: "+this.getColor()+" Filled: "+this.filled+" Date: "+this.getDateCreated());

}

}

class Rectangle extends GeometricObject{

private double width;

private double height;

public Rectangle(){

width = 1.0;

height = 1.0;

}

public Rectangle(double width, double height){

this.width = width;

this.height = height;

}

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

super(color,filled);

this.width = width;

this.height = height;

}

public double getWidth(){

return this.width;

}

public void setWidth(double width){

this.width = width;

}

public double getHeight(){

return this.height;

}

public void setHeight(double height){

this.height = height;

}

public double getArea(){

return this.width \* this.height;

}

public double getPerimeter(){

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

}

@Override

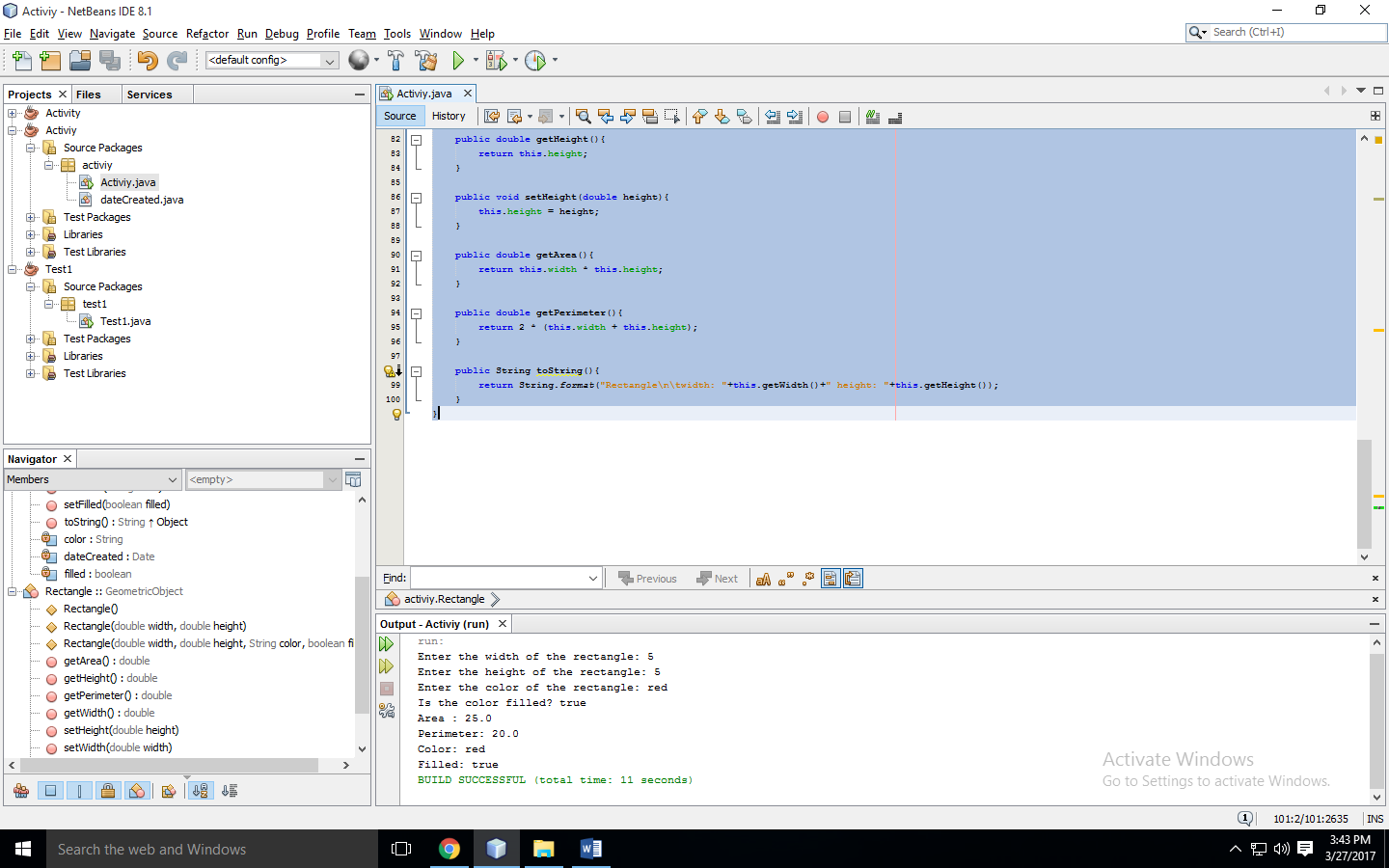
public String toString(){

return String.format("Rectangle\n\twidth: "+this.getWidth()+" height: "+this.getHeight());

}

}

**Output:**



**Task two:**

package activiy;

import java.util.\*;

public class Activiy {

public static void main(String[] args) {

Person p = new Person();

p.setName("Rizwan");

Student s = new Student();

Employee e = new Employee();

Faculty f = new Faculty();

Staff sf = new Staff();

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

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

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

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

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

}

}

class Person{

String name;

String address;

int contact;

String emailAddress;

public void setName(String name){

this.name = name;

}

public String getName(){

return name;

}

@Override

public String toString(){

return String.format("Class name: Person, Person name: %s", name);

}

}

class Student extends Person{

final String status;

Student(){

status = null;

}

Student(String status){

this.status = status;

}

public String toString(){

return String.format("Class name: Student, Person name: %s", super.getName());

}

}

class Employee extends Person{

int office;

double salary;

String dateHeired;

public String toString(){

return String.format("Class name: Employee, Person name: %s", super.getName());

}

}

class Faculty extends Employee{

double officeHours;

int rank;

public String toString(){

return String.format("Class name: Faculty, Person name: %s", super.getName());

}

}

class Staff extends Employee{

String title;

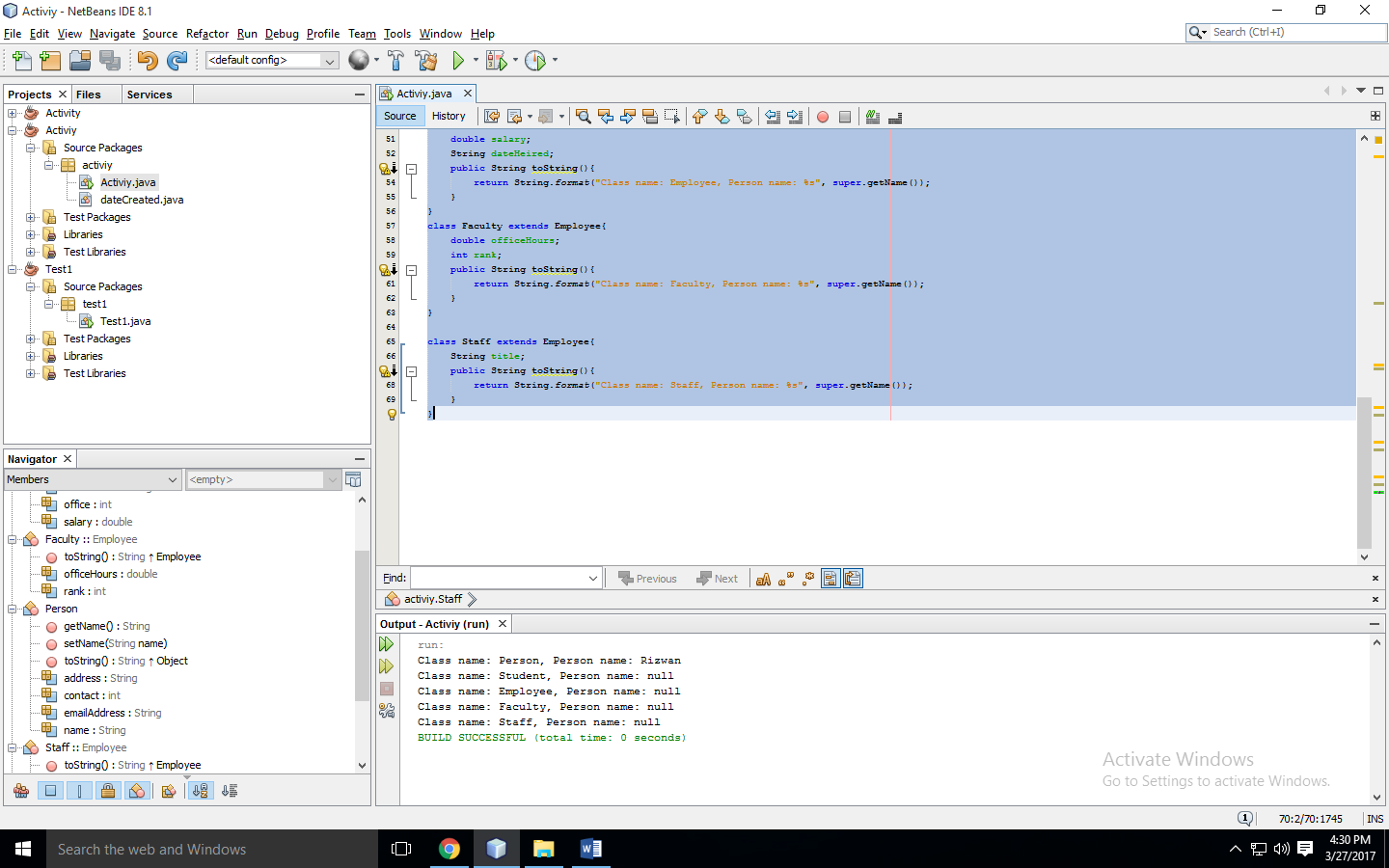
public String toString(){

return String.format("Class name: Staff, Person name: %s", super.getName());

}

}

**Output:**



**UML Diagram:**

|  |
| --- |
| Person |
| name : string  Address: string  Contact : int  Email : string |
| toString(): string  setName(name: string)  getName(): string |

|  |
| --- |
| Faculty |
| Officehours : double  Rank : int |
| toString(): string |

|  |
| --- |
| Employee |
| Office : int  Salary: double  dateHired: striing |
| toString(): String |

|  |
| --- |
| Student |
| <<constructor>> Student()  <<constructor>>Student(status:String) |
| toString(): string |

|  |
| --- |
| Staff |
| Title : string |
| toString(): string |