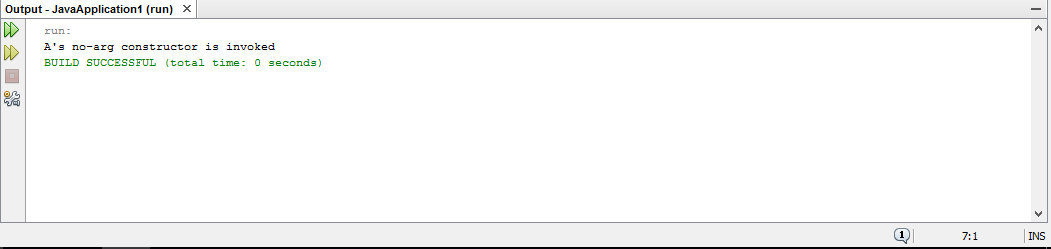
**ABDUL BASIT**

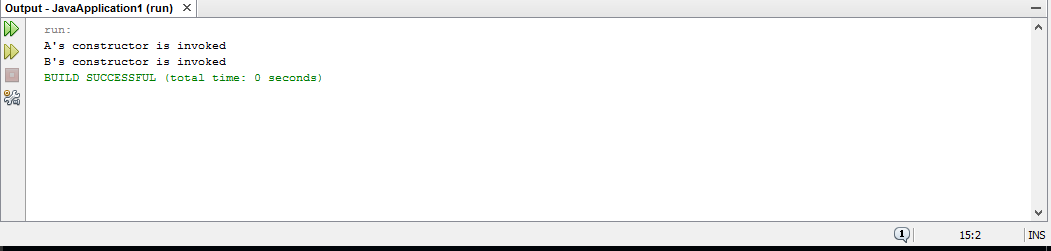
**00000193227**

**BSCS\_6\_CS**

**ACTIVITY ONE**

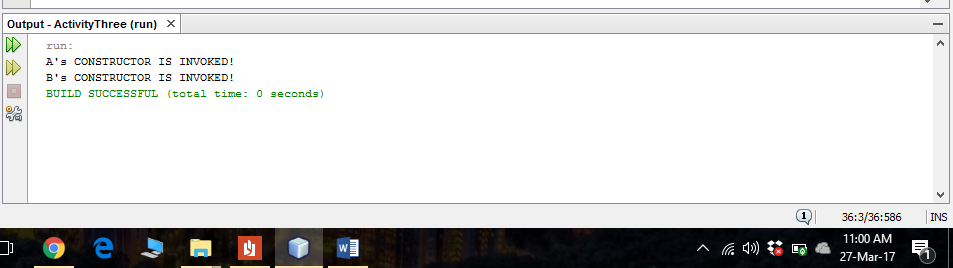
The no args constructor of the super class is called and because of this the print statement works because we can invoke the constructors of super class.

**ACTIVITY TWO**

When an object is created its base constructor called automatically.

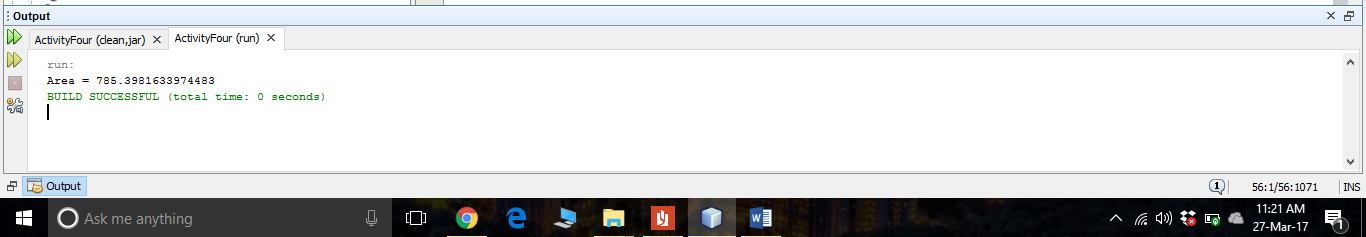
**ACTRIVITY\_3**

Creating default constructor in base class removes error.



**ACTIVITY 4**

1. CREATING DAFAULT CONSTRUCTOR IN BASE CLASS IS BASIC CORRECTION.
2. CHANGING GET AREA METHOD NAME IN CLASS ‘B’ IS SECOND CORRECTION.



**TASK\_1**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*///IMPORTING LIBRARY

import java.util.Date;

import java.util.Scanner;

/\*\*

\*

\* @author abasit.bscs16seecs

\*///START MAIN CLASS

public class GeometricObject {

//FIELDS

private String color;

private boolean filled;

//CONSTRUCTORS

public GeometricObject(){}

public GeometricObject(String color,boolean filled){

this.color=color;

this.filled=filled;

}

//GETTER METHOD FOR COLOR

public String GetColor(){

return color;

}

//SETTER METHOD FOR COLOR

public void setColor(String color){

this.color=color;

}

//FUNCTION FOR CHECKING STATUS

public boolean isFilled(){

return filled;

}

//FILLED FUNCTION

public void setFilled(boolean filled){

this.filled=filled;

}

//FUNCTION FOR GETTING DATE

public String getDateCreated(){

Date d = new Date();

return d.toString();

}

@Override//USING OOVERRIDE FOR NEXT CLASS

public String toString(){

return String.format("");

}

//METHOD FOR GETTING COLOR

public String getColor(){

return color;

}

//METHOD FOR GETTING FILL STATUS

public boolean getFilled(){

return filled;

}

}

//INHERITING CLASS FROM SUPER CLASS GEOMETRIC OBJECT

class Circle extends GeometricObject{

//INSTANCE VARIABLES

private double radius;

//CONSTRUCTORS

public Circle(){}

public Circle(double radius){}

public Circle(double radilus,String color,boolean filled){

super(color,filled);//CALLING SUPER CLASS

setRadius(radius);//CALLING SETTER FOR RADIUS

}

//GETTER AND SETTER METHODS

public double getRadius(){

return radius;

}

public void setRadius(double radius){

this.radius=radius;

}

public double getArea(){

return 3.1416\*radius\*radius;

}

public double getPerimeter(){

return 2\*3.1416\*radius;

}

public double getDiameter(){

return 2\*radius;

}

}

//CREATING INHERITED RECTANGLE CLASS FROM SUPER CLASS GEOMETRIC OBJECT

class Rectangle extends GeometricObject{

private double width;

private double height;

//CONSTRUCTORS

public Rectangle(){}

public Rectangle(double width,double height){}

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

super(color,filled);//CALLING CONSTRUCTOR OF SUPER CLASS

setWidth(width);//CALLING SETTER

setHeight(height);//CALLING SETTER

}

//SETTER AND GETTER FUNCTIONS

public void setHeight(double height){

this.height=height;

}

public double getHeight(){

return height;

}

public void setWidth(double width){

this.width=width;

}

public double getWidth(){

return width;

}

public double getArea(){

return width\*height;

}

public double getPerimeter(){

return 2\*(height+width);

}

//OVERRIDING

@Override

public String toString(){

return String.format("\nRectangle:\nWidth:%.2f\nHeight:%.2f \nColor:%s \nStatus:%b",height,width,getColor(),getFilled());

}

}

//MAIN CLASS BEGINS FOR TESTING

class Test{

//MAIN FUNCTION

public static void main(String args[]){

//CREATING OBJECT FOR INPUT

Scanner input=new Scanner(System.in);

//GETTING WIDTH

System.out.print("Enter width:");

double width=input.nextDouble();

//GETTING HEIGHT

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

double height=input.nextDouble();

//GETTING COLOR

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

String color=input.next();

//GETTING STATUS OF RECTANGLE

System.out.print("Enter status:");

boolean filled=input.nextBoolean();

//CREATING OBJECT FROM RECTANGLE CLASS

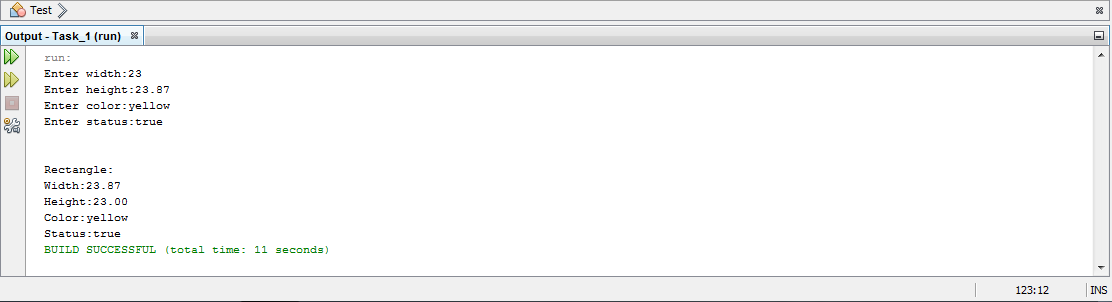
Rectangle R=new Rectangle(width,height,color,filled);

//CALLING TO STRIING FUNCTION

System.out.printf("%s\n",R.toString());

}//END MAIN

}//END TEST CLASS

**OUTPUT TASK\_1**

**TASK\_2**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Abdul Basit

\*/

package task2;

class Person

{

public String name;

private String address;

private String email;

private long number;

Person(){}

Person(String name,String address,String email,long number)

{

setName(name);

setAddress(address);

setEmail(email);

setNumber(number);

}

public void setName(String name)

{

this.name=name;

}

String getName()

{

return name;

}

public void setAddress(String address)

{

this.address=address;

}

String getAddress()

{

return address;

}

public void setEmail(String email)

{

this.email=email;

}

String getEmail()

{

return email;

}

public void setNumber(long number)

{

this.number=number;

}

public long getNumber()

{

return number;

}

public String toString()

{

return "CLASS: PERSON\n NAME:" + name;

}

}

//CLASS STUDENT INHERITING FROM PERSON CLASS

class Student extends Person

{

final String status = "FRESHMAN";

Student(){}

public String toString()

{

return "CLASS: PERSON\n NAME:" + name;

}

}

class Employee extends Person

{

private String office;

private long salary;

private String dateHired;

public void setOffice(String office)

{

this.office=office;

}

String getOffice()

{

return office;

}

public void setSalary(long salary)

{

this.salary=salary;

}

public long getSalary()

{

return salary;

}

public void setDateHired(String DateHired)

{

this.dateHired=DateHired;

}

public String getDateHired()

{

return dateHired;

}

public String toString()

{

return "CLASS: EMPLOYEE\n NAME:" + name;

}

}

//FACULTY CLASS INHERITING FROM EMPLOYEE CLASS

class Faculty extends Employee

{

private int officeHours;

private String rank;

Faculty(){}

public Faculty(int officeHours,String rank)

{

this.officeHours=officeHours;

this.rank=rank;

}

public void setOfficeHours(int hours)

{

this.officeHours=hours;

}

public int getOfficeHours()

{

return officeHours;

}

public void setRank(String rank)

{

this.rank=rank;

}

public String getRank()

{

return rank;

}

public String toString()

{

return "CLASS: FACULTY\n NAME:" + name;

}

}

//CLASS STAFF INHERITED FROM CLASS EMPLOYEE

class Staff extends Employee

{

private String title;

public Staff(){}

public Staff(String title)

{

this.title=title;

}

public void setTitle(String title)

{

this.title=title;

}

public String getTitle()

{

return title;

}

public String toString()

{

return "CLASS: STAFF \n NAME:" + name;

}

}

public class Test {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args)

{

Person O1=new Person("Basit","Lahore","abasit.bscs16seecs@seecs.edu.pk",193227);

System.out.printf("%s",O1);

Student O2=new Student();

System.out.printf("%s",O2);

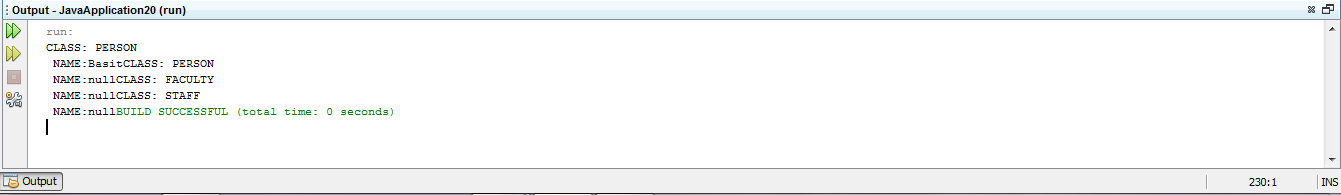
Faculty O3=new Faculty(12,"BOSS");

System.out.printf("%s",O3);

Staff O4 =new Staff("ADMIN EVENTS");

System.out.printf("%s",O4);

}

}

**OUTPUT TASK\_2**