INSTITUTE OF TECHNOLOGY AND MANAGEMENT
SKILLS UNIVERSITY,
KHARGHAR, NAVI MUMBAI

Java

Prepared by:

Name of Student : Chaitanya Dalvi

Roll No: 19

Batch: 2023-27

Dept. of CSE

Name of Student: Chaitanya Dalvi

Roll Number: 19

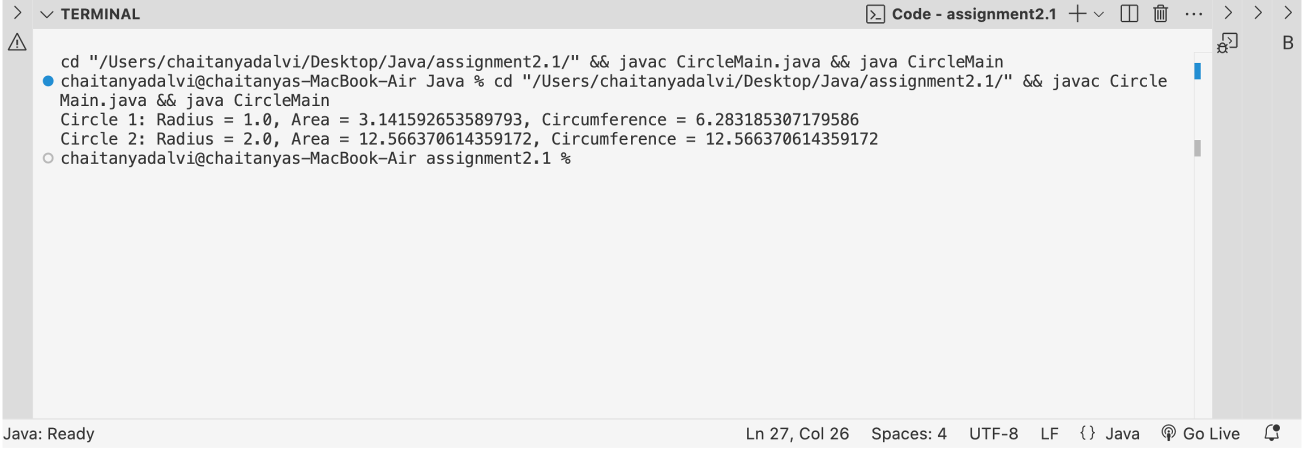
Experiment No: 1

Title: WAP to create a class called **Circle.** It contains:

Code:

```
class Circle {
    private double radius;

    Circle() {
        radius = 1.0;
    }

    Circle(double radius) {
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public double calculateArea() {
        return Math.PI * radius * radius;
    }

    public double calculateCircumference() {
        return 2 * Math.PI * radius;
    }
}

public class CircleMain {
    public static void main(String[] args) {
        Circle c1 = new Circle();
        Circle c2 = new Circle(2.0);
        System.out.println("Circle 1: Radius = " + c1.getRadius() +
", Area = " + c1.calculateArea()
                + ", Circumference = " +
c1.calculateCircumference());
        System.out.println("Circle 2: Radius = " + c2.getRadius() +
", Area = " + c2.calculateArea()
                + ", Circumference = " +
c2.calculateCircumference());
    }
}
```

Output: (screenshot)



```
 cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac CircleMain.java && java CircleMain
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Circle
Main.java && java CircleMain
 Circle 1: Radius = 1.0, Area = 3.141592653589793, Circumference = 6.283185307179586
 Circle 2: Radius = 2.0, Area = 12.566370614359172, Circumference = 12.566370614359172
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

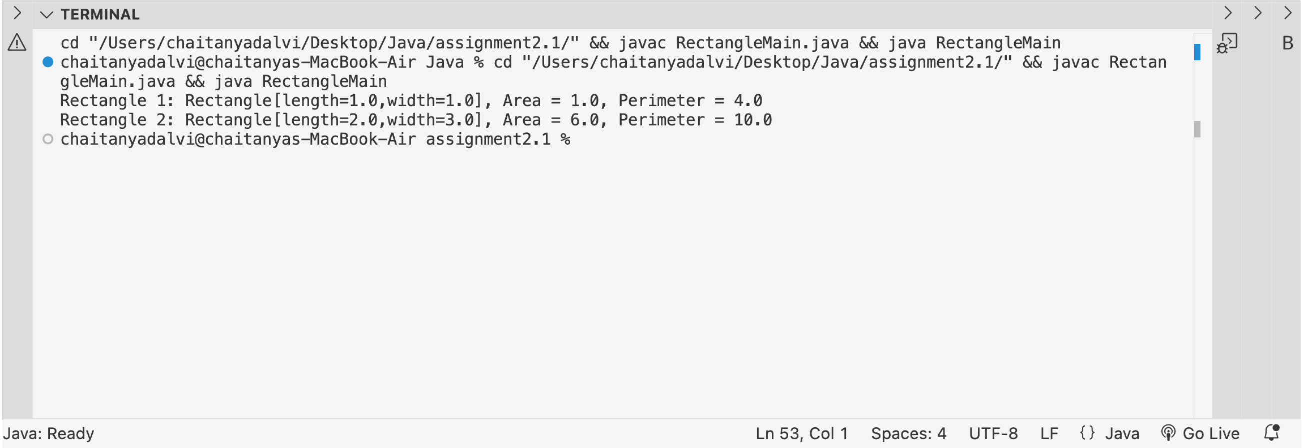
Experiment No: 2

Title: A class called Rectangle, which models a rectangle with a length and a width (in float), is designed as shown in the following class diagram. Write the Rectangle class as per UML diagram.

Code:

```
class Rectangle {
    private float length, width;

    Rectangle() {
        length = 1.0f;
        width = 1.0f;
    }

    Rectangle(float length, float width) {
        this.length = length;
        this.width = width;
    }

    public float getLength() {
        return length;
    }

    public void setLength(float length) {
        this.length = length;
    }

    public float getWidth() {
        return width;
    }

    public void setWidth(float width) {
        this.width = width;
    }

    public double getArea() {
        return length * width;
    }

    public double getPerimeter() {
        return 2 * (length + width);
    }

    public String toString() {
        return "Rectangle[length=" + length + ",width=" + width +
"]";
    }
}

public class RectangleMain {
    public static void main(String[] args) {
        Rectangle r1 = new Rectangle();
        Rectangle r2 = new Rectangle(2.0f, 3.0f);
        System.out.println("Rectangle 1: " + r1 + ", Area = " +
r1.getArea() + ", Perimeter = " + r1.getPerimeter());
        System.out.println("Rectangle 2: " + r2 + ", Area = " +
r2.getArea() + ", Perimeter = " + r2.getPerimeter());
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac RectangleMain.java && java RectangleMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Rectan
gleMain.java && java RectangleMain
Rectangle 1: Rectangle[length=1.0,width=1.0], Area = 1.0, Perimeter = 4.0
Rectangle 2: Rectangle[length=2.0,width=3.0], Area = 6.0, Perimeter = 10.0
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

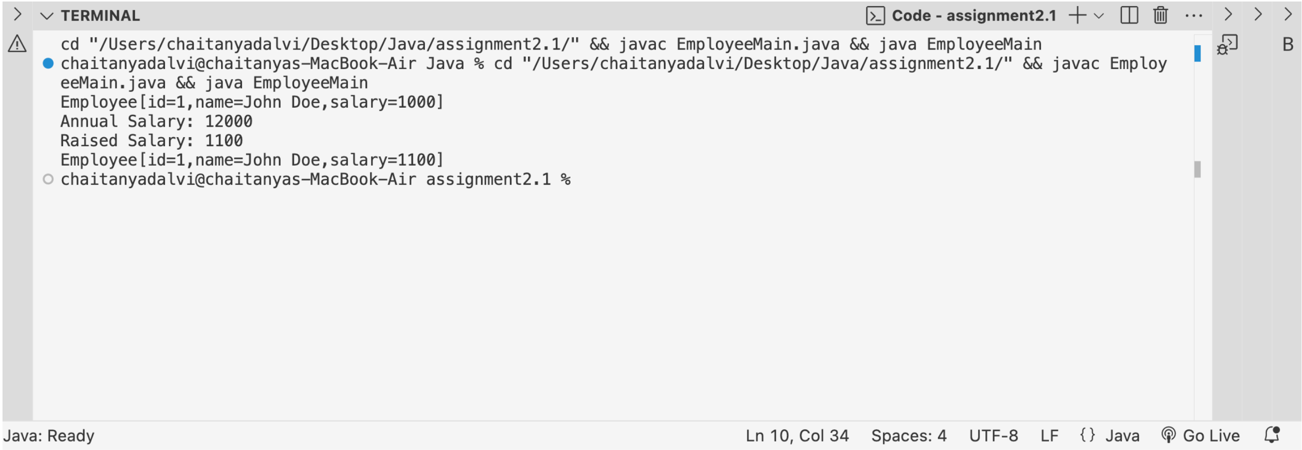
Experiment No: 3

Title: A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary(percent) increases the salary by the given percentage. Write the Employee class and its driver class.

Code:

```
class Employee {
    private int id, salary;
    private String firstName, lastName;

    Employee(int id, String firstName, String lastName, int salary)
{
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.salary = salary;
    }

    public int getId() {
        return id;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getName() {
        return firstName + " " + lastName;
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }

    public int getAnnualSalary() {
        return salary * 12;
    }

    public int raiseSalary(int percent) {
        salary += salary * percent / 100;
        return salary;
    }

    public String toString() {
        return "Employee[id=" + id + ",name=" + getName() +
",salary=" + salary + "]";
    }
}

public class EmployeeMain {
    public static void main(String[] args) {
        Employee e1 = new Employee(1, "John", "Doe", 1000);
        System.out.println(e1);
        System.out.println("Annual Salary: " +
e1.getAnnualSalary());
        System.out.println("Raised Salary: " + e1.raiseSalary(10));
        System.out.println(e1);
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac EmployeeMain.java && java EmployeeMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Employ
eeMain.java && java EmployeeMain
Employee[id=1,name=John Doe,salary=1000]
Annual Salary: 12000
Raised Salary: 1100
Employee[id=1,name=John Doe,salary=1100]
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

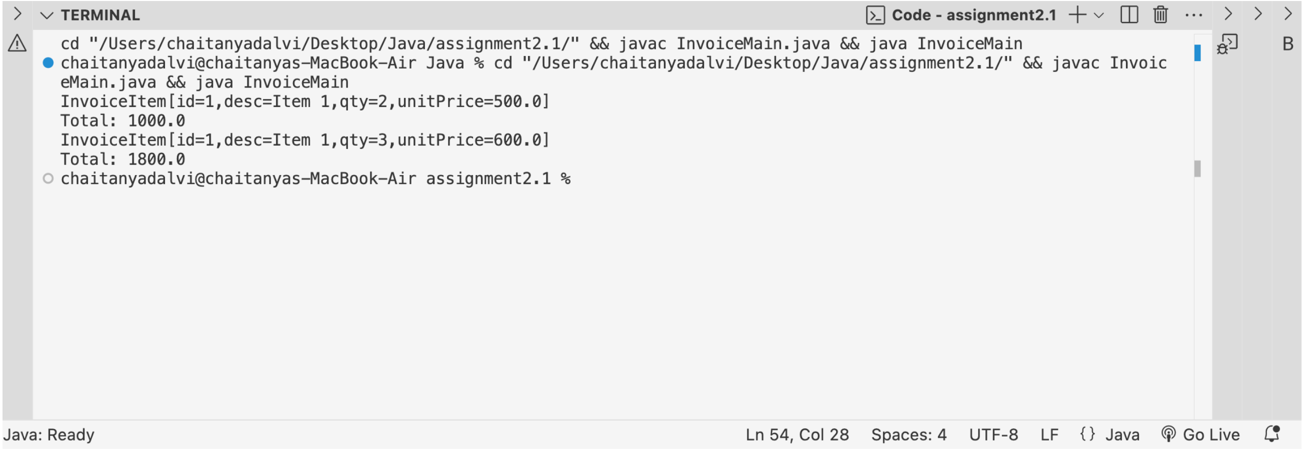
Experiment No: 4

Title: A class called InvoiceItem, which models an item of an invoice, with ID, description, quantity and unit price, is designed as shown in the following class diagram. It has a method getTotal which calculates total value (total=quantity*unit price). Write the InvoiceItem class and it’s driver class.

Code:

```
class InvoiceItem {
    private String id, desc;
    private int qty;
    private double unitPrice;

    InvoiceItem(String id, String desc, int qty, double unitPrice)
{
        this.id = id;
        this.desc = desc;
        this.qty = qty;
        this.unitPrice = unitPrice;
    }

    public String getId() {
        return id;
    }

    public String getDesc() {
        return desc;
    }

    public int getQty() {
        return qty;
    }

    public void setQty(int qty) {
        this.qty = qty;
    }

    public double getUnitPrice() {
        return unitPrice;
    }

    public void setUnitPrice(double unitPrice) {
        this.unitPrice = unitPrice;
    }

    public double getTotal() {
        return qty * unitPrice;
    }

    public String toString() {
        return "InvoiceItem[id=" + id + ",desc=" + desc + ",qty=" +
qty + ",unitPrice=" + unitPrice + "]";
    }
}

public class InvoiceMain {
    public static void main(String[] args) {
        InvoiceItem i1 = new InvoiceItem("1", "Item 1", 2, 500.0);
        System.out.println(i1);
        System.out.println("Total: " + i1.getTotal());
        i1.setQty(3);
        i1.setUnitPrice(600.0);
        System.out.println(i1);
        System.out.println("Total: " + i1.getTotal());
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac InvoiceMain.java && java InvoiceMain
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Invoic
eMain.java && java InvoiceMain
InvoiceItem[id=1,desc=Item 1,qty=2,unitPrice=500.0]
Total: 1000.0
InvoiceItem[id=1,desc=Item 1,qty=3,unitPrice=600.0]
Total: 1800.0
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

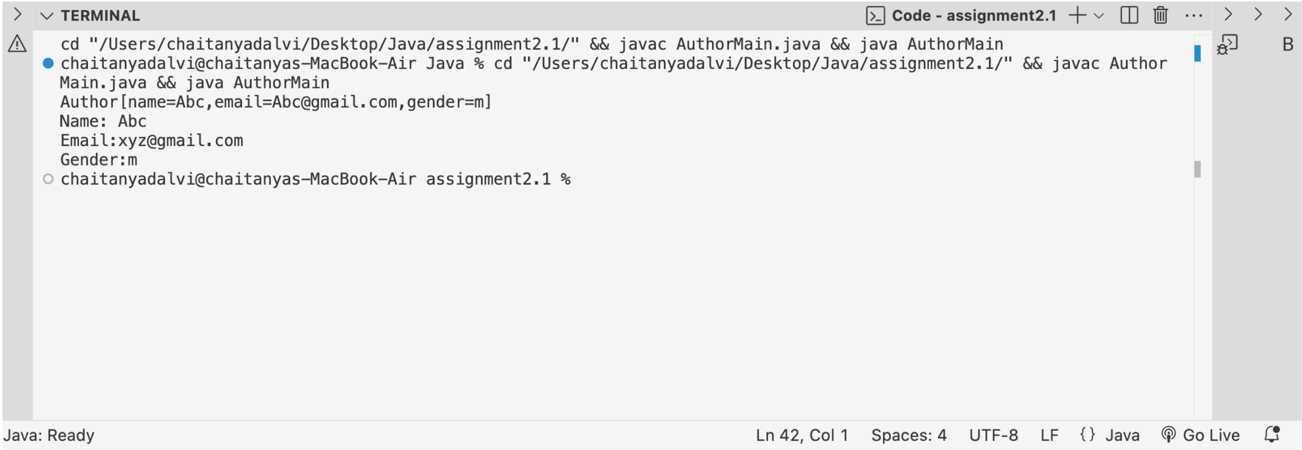
Experiment No: 5

Title: A class called Author (as shown in the class diagram) is designed to model a book's author.

Code:

```
class Author {
    private String name, email;
    private char gender;

    Author(String name, String email, char gender) {
        this.name = name;
        this.email = email;
        this.gender = gender;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public char getGender() {
        return gender;
    }
    public String toString() {
        return "Author[name=" + name + ",email=" + email +
",gender=" + gender + "]";
    }
}

public class AuthorMain {
    public static void main(String[] args) {
        Author a1 = new Author("Abc", "Abc@gmail.com", 'm');
        System.out.println(a1);
        a1.setEmail("xyz@gmail.com");
        System.out.println("Name: " + a1.getName() + "\nEmail:" +
a1.getEmail() + "\nGender:" + a1.getGender());
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac AuthorMain.java && java AuthorMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Author
Main.java && java AuthorMain
Author[name=Abc,email=Abc@gmail.com,gender=m]
Name: Abc
Email:xyz@gmail.com
Gender:m
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 6

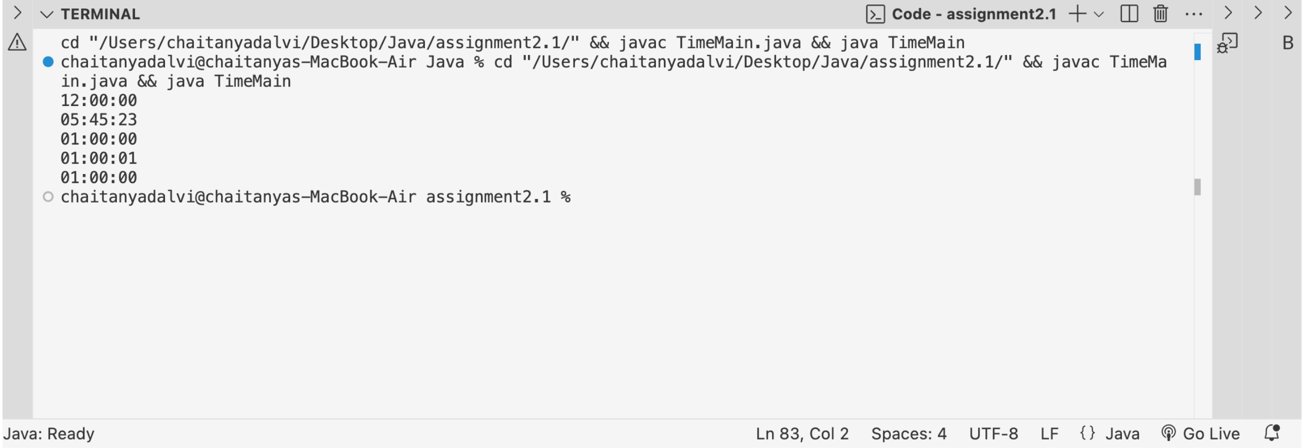
Title: WAP to create a class time having default constructor, parameterized constructor whose specifications

Code:

```
class Time {
    private int hr, min, sec;

    Time() {
        this.hr = 12;
        this.min = 0;
        this.sec = 0;
    }

    Time(int hr, int min, int sec) {
        this.hr = hr;
        this.min = min;
        this.sec = sec;
        normalizeTime();
    }

    Time(int secondsSinceMidnight) {
        setClock(secondsSinceMidnight);
    }

    public void setClock(int secondsSinceMidnight) {
        this.hr = secondsSinceMidnight / 3600;
        this.min = (secondsSinceMidnight % 3600) / 60;
        this.sec = secondsSinceMidnight % 60;
        normalizeTime();
```

```java
    }

    public void tick() {
        this.sec++;
        normalizeTime();
    }

    public void tickDown() {
        this.sec--;
        normalizeTime();
    }

    private void normalizeTime() {
        if (this.sec >= 60) {
            this.min += this.sec / 60;
            this.sec %= 60;
        }
        if (this.min >= 60) {
            this.hr += this.min / 60;
            this.min %= 60;
        }
        if (this.sec < 0) {
            this.min--;
            this.sec += 60;
        }
        if (this.min < 0) {
            this.hr--;
            this.min += 60;
        }
        if (this.hr >= 24) {
            this.hr %= 24;
        }
        if (this.hr < 0) {
            this.hr += 24;
        }
    }

    public void displayTime() {
        System.out.printf("%02d:%02d:%02d\n", hr, min, sec);
    }
}

public class TimeMain {
    public static void main(String[] args) {
        Time t1 = new Time();
        t1.displayTime();

        Time t2 = new Time(5, 45, 23);
        t2.displayTime();
        Time t3 = new Time(3600);
        t3.displayTime();
        t3.tick();
        t3.displayTime();
        t3.tickDown();
        t3.displayTime();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac TimeMain.java && java TimeMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac TimeMa
in.java && java TimeMain
12:00:00
05:45:23
01:00:00
01:00:01
01:00:00
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

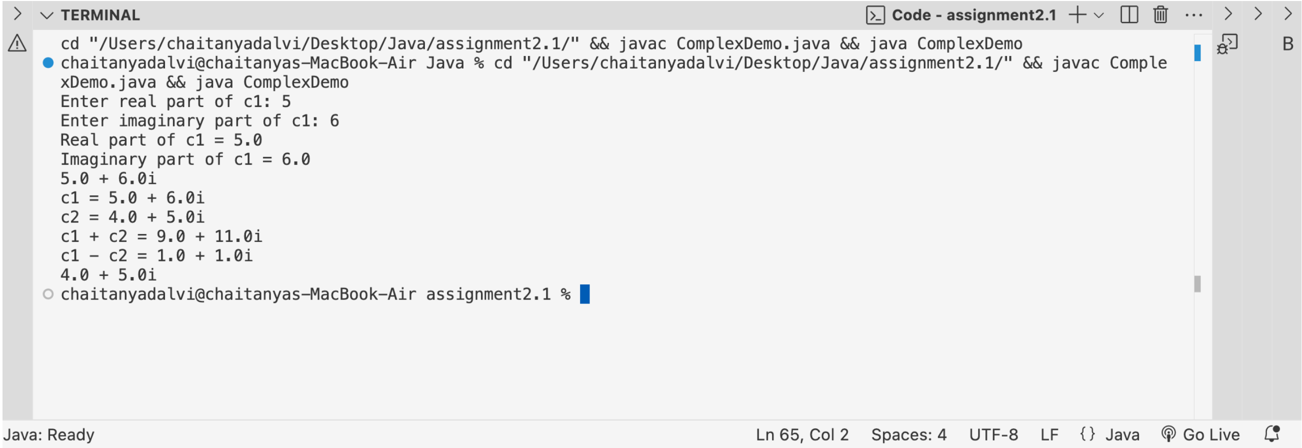
Experiment No: 7

Title: Write a Java class Complex for dealing with complex number.

Code:

```java
import java.util.Scanner;

class Complex {
    private double real, imag;

    public Complex() {
        real = imag = 0;
    }

    public Complex(double real, double imag) {
        this.real = real;
        this.imag = imag;
    }

    public double getReal() {
        return real;
    }

    public double getImag() {
        return imag;
    }

    public void setReal(double real) {
        this.real = real;
    }

    public void setImag(double imag) {
        this.imag = imag;
    }

    public String toString() {
        return real + " + " + imag + "i";
    }
    public Complex add(Complex c) {
        return new Complex(real + c.real, imag + c.imag);
    }

    public Complex subtract(Complex c) {
        return new Complex(real - c.real, imag - c.imag);
    }
}

public class ComplexDemo {
    public static void main(String[] args) {
        Complex c1 = new Complex();
        Complex c2 = new Complex(4, 5);
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter real part of c1: ");
        c1.setReal(sc.nextDouble());
        System.out.print("Enter imaginary part of c1: ");
        c1.setImag(sc.nextDouble());
        System.out.println("Real part of c1 = " + c1.getReal());
        System.out.println("Imaginary part of c1 = " +
c1.getImag());
        System.out.println(c1);
        System.out.println("c1 = " + c1);
        System.out.println("c2 = " + c2);
        System.out.println("c1 + c2 = " + c1.add(c2));
        System.out.println("c1 - c2 = " + c1.subtract(c2));
        System.out.println(c2);
        sc.close();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac ComplexDemo.java && java ComplexDemo
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Comple
xDemo.java && java ComplexDemo
Enter real part of c1: 5
Enter imaginary part of c1: 6
Real part of c1 = 5.0
Imaginary part of c1 = 6.0
5.0 + 6.0i
c1 = 5.0 + 6.0i
c2 = 4.0 + 5.0i
c1 + c2 = 9.0 + 11.0i
c1 - c2 = 1.0 + 1.0i
4.0 + 5.0i
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

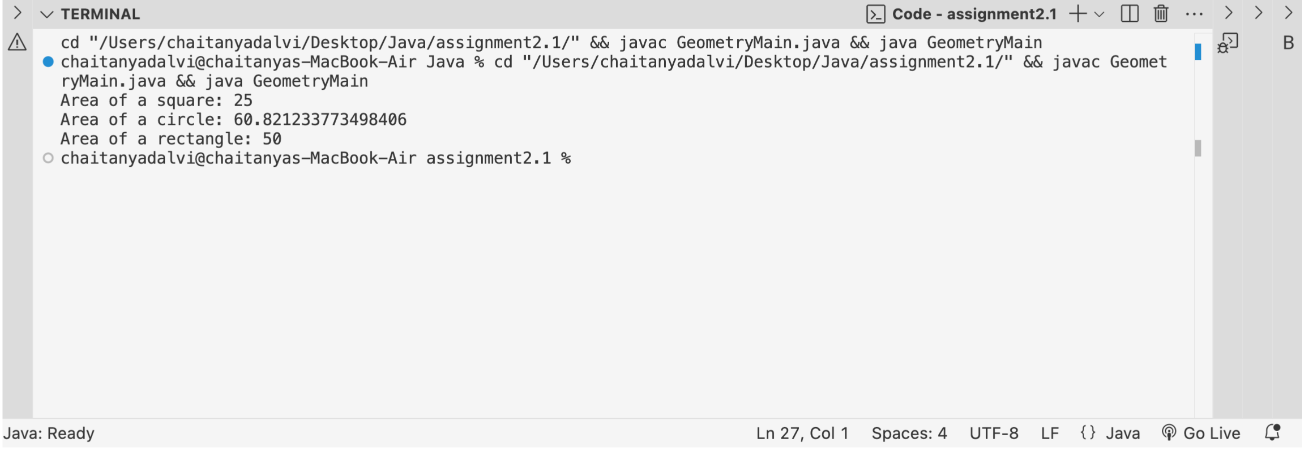
Roll Number: 19

Experiment No: 8

Title: Create a Geometry class and overload calculateArea method for square, circle, and rectangle.

```java
class Geometry {
    public int calculateArea(int s) {
        return s * s;
    }

    public double calculateArea(double r) {
        return r * r * Math.PI;
    }

    public int calculateArea(int l, int b) {
        return l * b;
    }
}

public class GeometryMain {
    public static void main(String[] args) {
        Geometry square = new Geometry();
        Geometry circle = new Geometry();
        Geometry rectangle = new Geometry();
        System.out.println("Area of a square: " +
square.calculateArea(5));
        System.out.println("Area of a circle: " +
circle.calculateArea(4.4));
        System.out.println("Area of a rectangle: " +
rectangle.calculateArea(5, 10));
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac GeometryMain.java && java GeometryMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Geomet
ryMain.java && java GeometryMain
Area of a square: 25
Area of a circle: 60.821233773498406
Area of a rectangle: 50
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 9

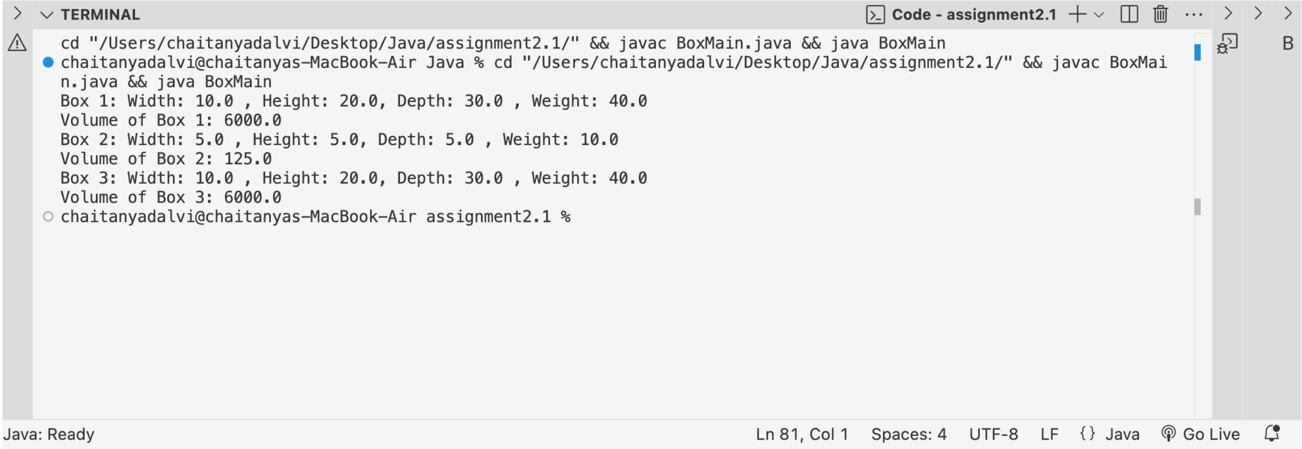
Title: WAP to implement Box class (Hint: Refer Java PPT). Inherit Box class in BoxWt

Code:

```java
class Box {
    private double w, h, d;

    Box() {
        w = h = d = 0;
    }

    Box(double w, double h, double d) {
        this.w = w;
        this.h = h;
        this.d = d;
    }

    Box(double a) {
        w = h = d = a;
    }

    Box(Box obj) {
        w = obj.w;
        h = obj.h;
        d = obj.d;
    }

    public double volume() {
        return w * h * d;
    }

    public String toString() {
        return ("Width: " + w + " , Height: " + h + ", Depth: " + d);
    }
}
```

```java
class BoxWt extends Box {
    private double wt;

    BoxWt() {
        super();
        wt = 0;
    }

    BoxWt(double w, double h, double d, double wt) {
        super(w, h, d);
        this.wt = wt;
    }

    BoxWt(double a, double wt) {
        super(a);
        this.wt = wt;
    }

    BoxWt(BoxWt obj) {
        super(obj);
        wt = obj.wt;
    }

    public String print_BoxWt() {
        return ("Weight: " + wt);
    }

    public String toString() {
        return (super.toString() + " , " +  print_BoxWt());
    }
}

public class BoxMain {
    public static void main(String[] args) {
        BoxWt obj1 = new BoxWt(10, 20, 30, 40);
        BoxWt obj2 = new BoxWt(5, 10);
        BoxWt obj3 = new BoxWt(obj1);

        System.out.println("Box 1: " + obj1);
        System.out.println("Volume of Box 1: " + obj1.volume());
        System.out.println("Box 2: " + obj2);
        System.out.println("Volume of Box 2: " + obj2.volume());
        System.out.println("Box 3: " + obj3);
        System.out.println("Volume of Box 3: " + obj3.volume());
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac BoxMain.java && java BoxMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac BoxMai
n.java && java BoxMain
Box 1: Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0
Volume of Box 1: 6000.0
Box 2: Width: 5.0 , Height: 5.0, Depth: 5.0 , Weight: 10.0
Volume of Box 2: 125.0
Box 3: Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0
Volume of Box 3: 6000.0
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 10

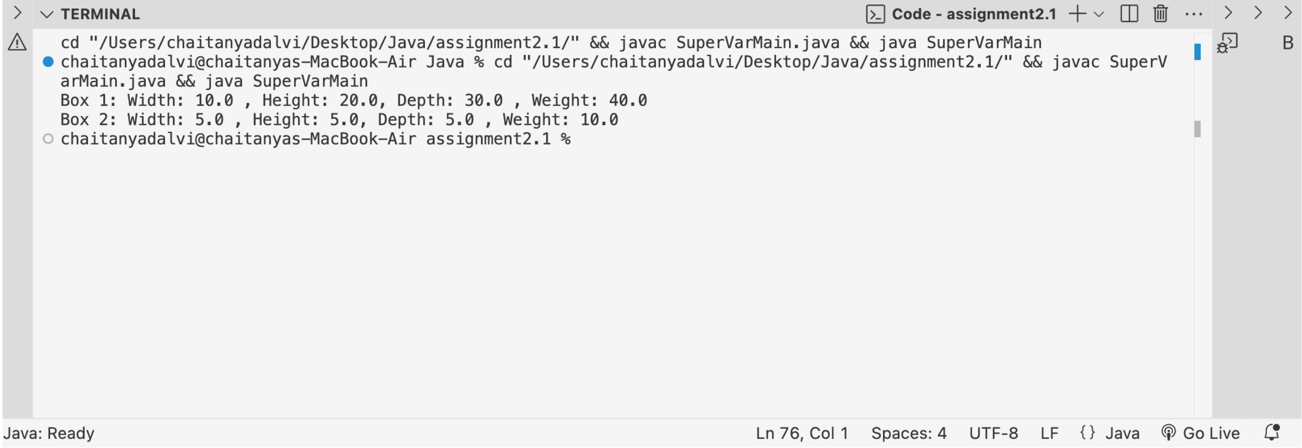
Title: WAP the demonstrate that super class variable can point to object of subclass.

Code:

```
class Box {
    private double w, h, d;

    Box() {
        w = h = d = 0;
    }

    Box(double w, double h, double d) {
        this.w = w;
        this.h = h;
        this.d = d;
    }

    Box(double a) {
        w = h = d = a;
    }

    Box(Box obj) {
        w = obj.w;
        h = obj.h;
        d = obj.d;
    }

    public double volume() {
        return w * h * d;
    }

    public String toString() {
        return ("Width: " + w + " , Height: " + h + ", Depth: " +
d);
    }
}
```

```java
class BoxWt extends Box {
    private double wt;

    BoxWt() {
        super();
        wt = 0;
    }

    BoxWt(double w, double h, double d, double wt) {
        super(w, h, d);
        this.wt = wt;
    }

    BoxWt(double a, double wt) {
        super(a);
        this.wt = wt;
    }

    BoxWt(BoxWt obj) {
        super(obj);
        wt = obj.wt;
    }

    public String print_BoxWt() {
        return ("Weight: " + wt);
    }

    public String toString() {
        return (super.toString() + " , " +  print_BoxWt());
    }
}

public class SuperVarMain {
    public static void main(String[] args) {
        Box obj1 = new BoxWt(10, 20, 30, 40);
        Box obj2 = new BoxWt(5, 10);

        System.out.println("Box 1: " + obj1);
        System.out.println("Box 2: " + obj2);
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SuperVarMain.java && java SuperVarMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SuperV
arMain.java && java SuperVarMain
Box 1: Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0
Box 2: Width: 5.0 , Height: 5.0, Depth: 5.0 , Weight: 10.0
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

Experiment No: 11

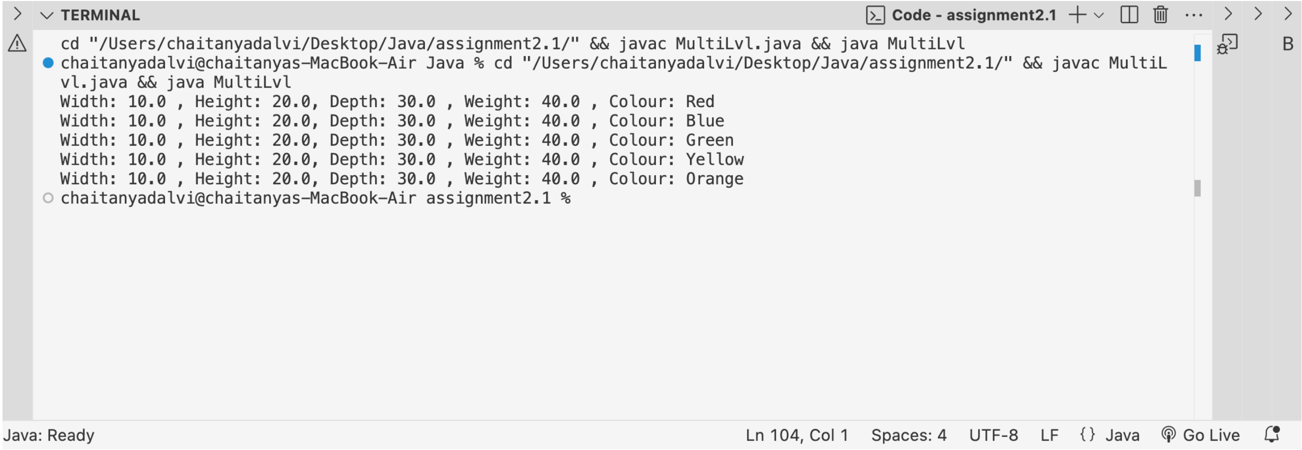
Title: WAP to demonstrate multilevel inheritance by creating a BoxColor class and inheriting BoxWt class. BoxColor class has an instance variable color of String type.

Code:

```
class Box {
    private double w, h, d;

    Box() {
        w = h = d = 0;
    }

    Box(double w, double h, double d) {
        this.w = w;
        this.h = h;
        this.d = d;
    }

    Box(double a) {
        w = h = d = a;
    }

    Box(Box obj) {
        w = obj.w;
        h = obj.h;
        d = obj.d;
    }

    public double volume() {
        return w * h * d;
    }

    public String toString() {
        return ("Width: " + w + " , Height: " + h + ", Depth: " + 
d);
    }
}

class BoxWt extends Box {
    private double wt;

    BoxWt() {
        super();
        wt = 0;
    }

    BoxWt(double w, double h, double d, double wt) {
        super(w, h, d);
        this.wt = wt;
    }

    BoxWt(double a, double wt) {
        super(a);
        this.wt = wt;
    }

    BoxWt(BoxWt obj) {
        super(obj);
        wt = obj.wt;
    }

    public String print_BoxWt() {
        return ("Weight: " + wt);
    }

    public String toString() {
        return (super.toString() + " , " +  print_BoxWt());
    }
}

class BoxColour extends BoxWt {
    private String colour;
    BoxColour(String colour) {
        super();
        this.colour = colour;
    }
    BoxColour(String colour, double w, double h, double d, double
wt) {
        super(w, h, d, wt);
        this.colour = colour;
    }
    BoxColour(String colour, double a, double wt) {
        super(a, wt);
        this.colour = colour;
    }
    BoxColour(String colour, BoxColour obj) {
        super(obj);
        this.colour = colour;
    }
    public String toString() {
        return (super.toString() + " , Colour: " + colour);
    }
```

```java
}

public class MultiLvl {
    public static void main(String[] args) {
        BoxColour box1 = new BoxColour("Red", 10, 20, 30, 40);
        BoxColour box2 = new BoxColour("Blue", 10, 20, 30, 40);
        BoxColour box3 = new BoxColour("Green", 10, 20, 30, 40);
        BoxColour box4 = new BoxColour("Yellow", 10, 20, 30, 40);
        BoxColour box5 = new BoxColour("Orange", 10, 20, 30, 40);
        System.out.println(box1);
        System.out.println(box2);
        System.out.println(box3);
        System.out.println(box4);
        System.out.println(box5);
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MultiLvl.java && java MultiLvl
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MultiL
vl.java && java MultiLvl
Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0 , Colour: Red
Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0 , Colour: Blue
Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0 , Colour: Green
Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0 , Colour: Yellow
Width: 10.0 , Height: 20.0, Depth: 30.0 , Weight: 40.0 , Colour: Orange
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

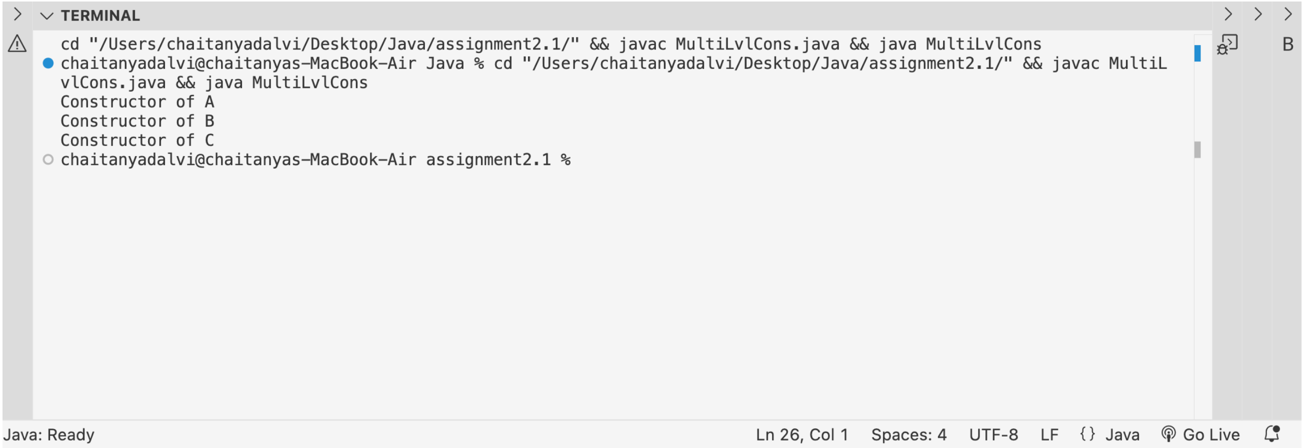
Experiment No: 12

Title: WAP to demonstrate the sequence of execution of constructors in multilevel inheritance.

Code:

```java
class A {
    A() {
        System.out.println("Constructor of A"); // This will be invoked first as it is the superclass
    }
}

class B extends A {
    B() {
        System.out.println("Constructor of B"); // This will be invoked second
    }
}

class C extends B {
    C() {
        System.out.println("Constructor of C"); // This will be invoked last
    }
}

public class MultiLvlCons {
    public static void main(String[] args) {
        C obj = new C();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MultiLvlCons.java && java MultiLvlCons
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MultiL
vlCons.java && java MultiLvlCons
Constructor of A
Constructor of B
Constructor of C
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

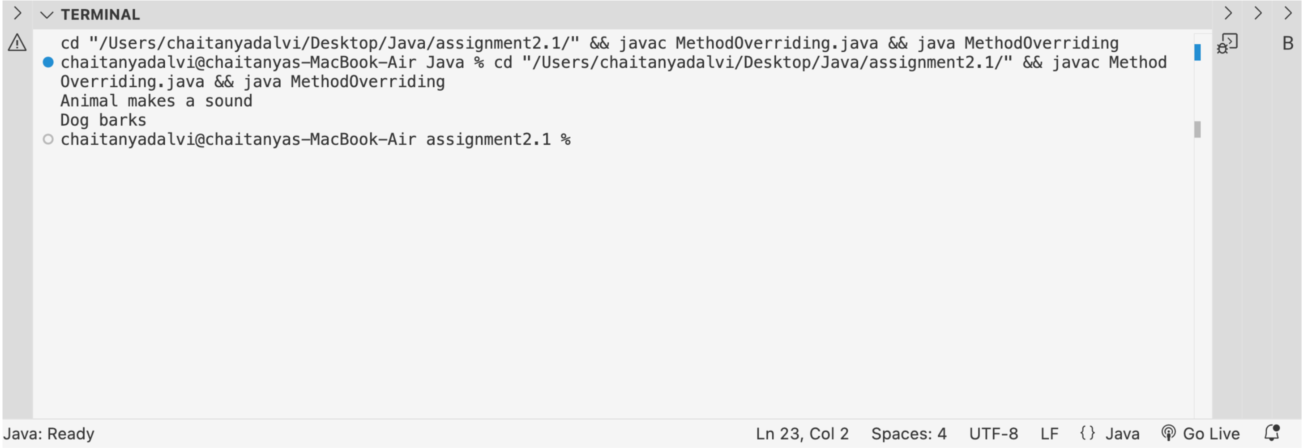
Experiment No: 13

Title: WAP to demonstrate the concept of method overriding.

Code:

```java
class Animal {
    public void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    public void sound() {
        System.out.println("Dog barks");
    }
}

public class MethodOverriding {
    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        myAnimal.sound(); // Calls the method from Animal class

        Dog myDog = new Dog();
        myDog.sound(); // Calls the overridden method from Dog
class
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MethodOverriding.java && java MethodOverriding
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Method
Overriding.java && java MethodOverriding
Animal makes a sound
Dog barks
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

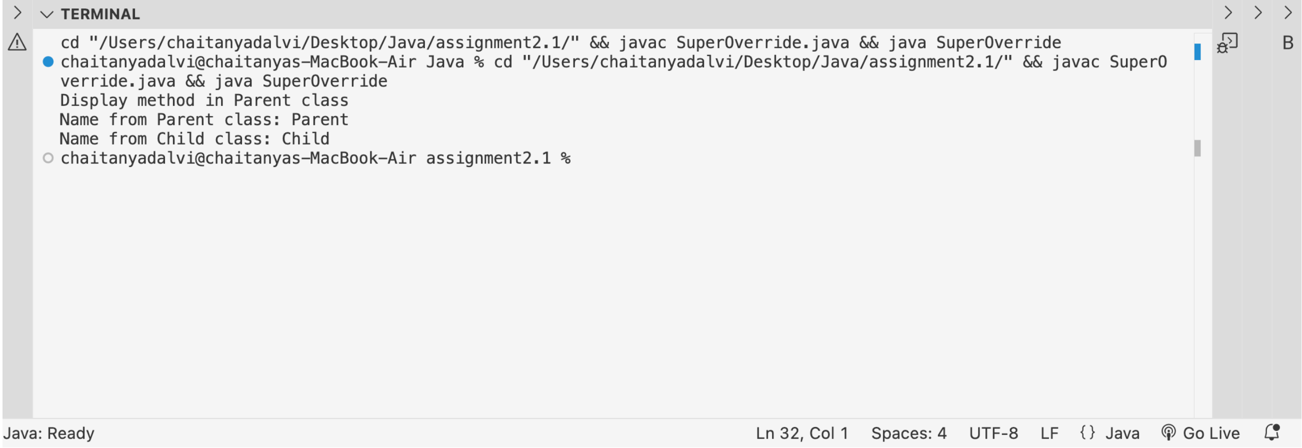
Experiment No: 14

Title: WAP to demonstrate the use of super keyword to call overridden methods and instance variables.

Code:

```
class Parent {
    String name = "Parent";

    public void display() {
        System.out.println("Display method in Parent class");
    }
}

// Derived class
class Child extends Parent {
    String name = "Child";

    public void display() {
        System.out.println("Display method in Child class");
    }

    public void show() {
        super.display();
        System.out.println("Name from Parent class: " +
super.name);
        System.out.println("Name from Child class: " + this.name);
    }
}

public class SuperOverride {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.show();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SuperOverride.java && java SuperOverride
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SuperO
verride.java && java SuperOverride
Display method in Parent class
Name from Parent class: Parent
Name from Child class: Child
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

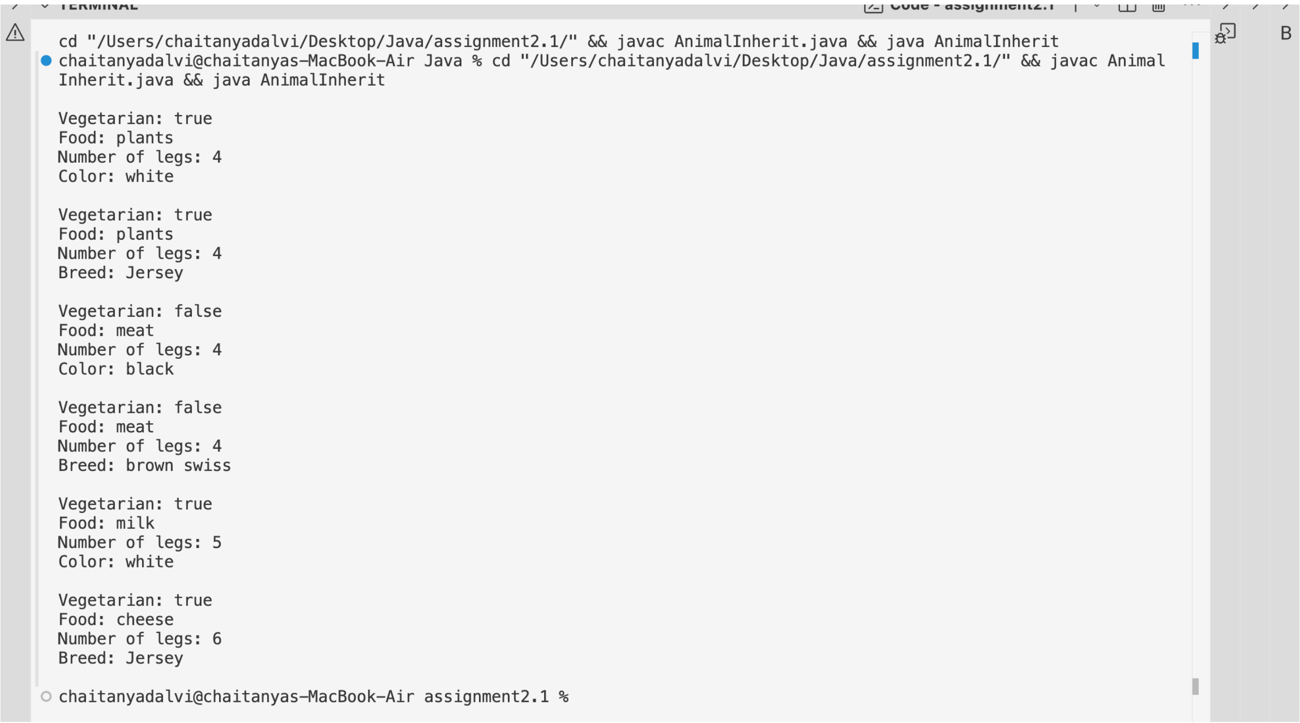
Experiment No: 15

Title: Create a class Animal

```
class Animal {

    boolean vegetarian;
    String food;
    int numOfLegs;

    Animal() {
        vegetarian = true;
        food = "plants";
        numOfLegs = 4;
    }

    Animal(boolean vegetarian, String food, int numOfLegs) {
        this.vegetarian = vegetarian;
        this.food = food;
        this.numOfLegs = numOfLegs;
    }

    public void setVegetarian(boolean vegetarian) {
        this.vegetarian = vegetarian;
    }

    public void setFood(String food) {
        this.food = food;
    }

    public void setNumOfLegs(int numOfLegs) {
        this.numOfLegs = numOfLegs;
    }

    public boolean isVegetarian() {
        return vegetarian;
    }

    public String getFood() {
        return food;
    }

    public int getNumOfLegs() {
        return numOfLegs;
    }

    public String toString() {
        return ("Vegetarian: " + vegetarian + "\nFood: " + food +
"\nNumber of legs: " + numOfLegs);
    }
}

class Cat extends Animal {
    String color;

    Cat() {
        super();
        color = "white";
    }

    Cat(boolean vegetarian, String food, int numOfLegs, String
color) {
        super(vegetarian, food, numOfLegs);
        this.color = color;
    }

    public String toString() {
        return (super.toString() + "\nColor: " + color);
    }
}

class Cow extends Animal {
    String breed;

    Cow() {
        super();
        breed = "Jersey";
    }

    Cow(boolean vegetarian, String food, int numOfLegs, String
breed) {
        super(vegetarian, food, numOfLegs);
        this.breed = breed;
    }

    public String toString() {
        return (super.toString() + "\nBreed: " + breed);
    }
}

public class AnimalInherit {
    public static void main(String[] args) {
```

```
    Cat c1 = new Cat();
    Cow cow1 = new Cow();
    System.out.println("\n" + c1 + "\n");
    System.out.println(cow1 + "\n");

    Cat c2 = new Cat(false, "meat", 4, "black");
    Cow cow2 = new Cow(false, "meat", 4, "brown swiss");
    System.out.println(c2.toString() + "\n");
    System.out.println(cow2.toString() + "\n");

    Cat c3 = new Cat();
    Cow cow3 = new Cow();
    c3.setVegetarian(true);
    c3.setFood("milk");
    c3.setNumOfLegs(5);
    c3.isVegetarian();
    c3.getFood();
    c3.getNumOfLegs();
    cow3.setVegetarian(true);
    cow3.setFood("cheese");
    cow3.setNumOfLegs(6);
    cow3.isVegetarian();
    cow3.getFood();
    cow3.getNumOfLegs();
    System.out.println(c3.toString() + "\n");
    System.out.println(cow3.toString() + "\n");
  }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac AnimalInherit.java && java AnimalInherit
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Animal
Inherit.java && java AnimalInherit

Vegetarian: true
Food: plants
Number of legs: 4
Color: white

Vegetarian: true
Food: plants
Number of legs: 4
Breed: Jersey

Vegetarian: false
Food: meat
Number of legs: 4
Color: black

Vegetarian: false
Food: meat
Number of legs: 4
Breed: brown swiss

Vegetarian: true
Food: milk
Number of legs: 5
Color: white

Vegetarian: true
Food: cheese
Number of legs: 6
Breed: Jersey

chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

Roll Number: 19

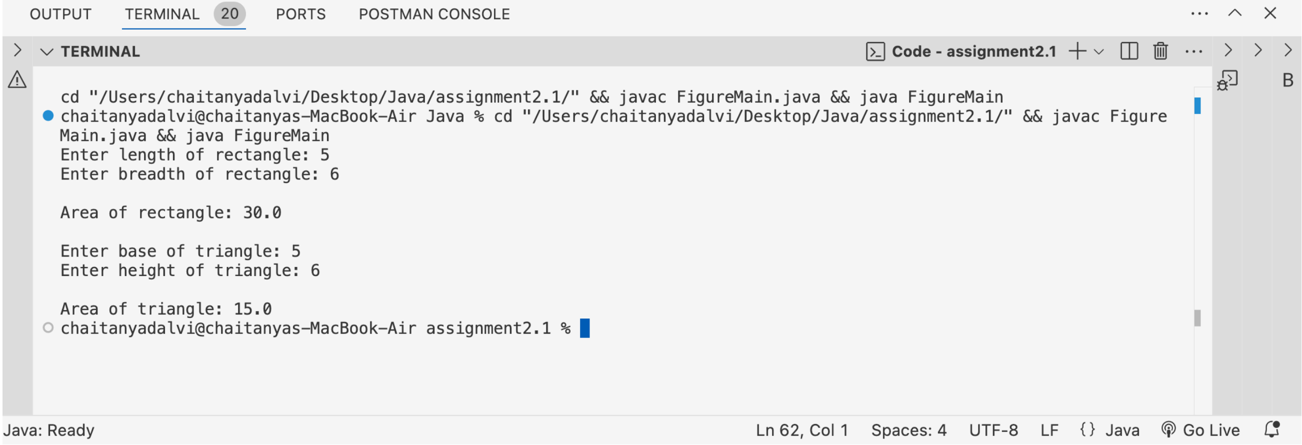
Experiment No: 16

Title: Create a figure class with instance variable dim1, dim2 and method as area(). Create two derived class rectangle and triangle with constructors defined for initializing instance variable. Override area in both derived classes. WAP to demonstrate dynamic method dispatch (Run time polymorphism).

```
import java.util.Scanner;

class Figure {
    double dim1, dim2;

    Figure(double a, double b) {
        dim1 = a;
        dim2 = b;
    }

    double area() {
        return 0;
    }
}

class Rectangle extends Figure {
    Rectangle(double a, double b) {
        super(a, b);
    }

    double area() {
        return dim1 * dim2;
    }
}

class Triangle extends Figure {
    Triangle(double a, double b) {
```

```java
        super(a, b);
    }

    double area() {
        return 0.5 * dim1 * dim2;
    }
}

public class FigureMain {
    public static void main(String[] args) {
        double a, b;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length of rectangle: ");
        a = sc.nextDouble();
        System.out.print("Enter breadth of rectangle: ");
        b = sc.nextDouble();
        Figure f;
        f = new Rectangle(a, b);
        System.out.println();
        System.out.println("Area of rectangle: " + f.area());
        System.out.println();

        System.out.print("Enter base of triangle: ");
        a = sc.nextDouble();
        System.out.print("Enter height of triangle: ");
        b = sc.nextDouble();
        f = new Triangle(a, b);
        System.out.println();
        System.out.println("Area of triangle: " + f.area());
        sc.close();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac FigureMain.java && java FigureMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Figure
Main.java && java FigureMain
Enter length of rectangle: 5
Enter breadth of rectangle: 6

Area of rectangle: 30.0

Enter base of triangle: 5
Enter height of triangle: 6

Area of triangle: 15.0
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

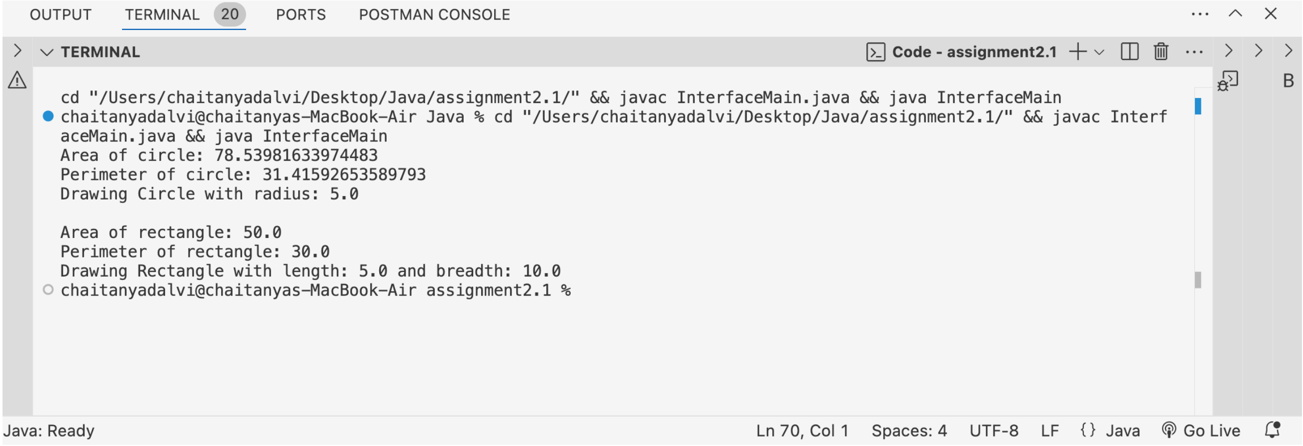
Roll Number: 19

Experiment No: 17

Title: Write a program to implement the concept of multiple inheritance through interface. Create a Figure and Draw interface. In Figure interface, define members PI, area(), perimeter(). In Draw interface, define members draw_shape(). Implement it in Circle class and Rectangle class.

```
interface Figure {
    double PI = Math.PI;

    double area();

    double perimeter();
}

interface Draw {
    void draw_shape(double a, double b);
}

class Circle implements Figure, Draw {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return PI * radius * radius;
    }

    public double perimeter() {
        return 2 * PI * radius;
    }

    public void draw_shape(double a, double b) {
        System.out.println("Drawing Circle with radius: " +
radius);
    }
}

class Rectangle implements Figure, Draw {
    double length, breadth;

    Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double area() {
        return length * breadth;
    }

    public double perimeter() {
        return 2 * (length + breadth);
    }

    public void draw_shape(double a, double b) {
        System.out.println("Drawing Rectangle with length: " +
length + " and breadth: " + breadth);
    }
}

public class InterfaceMain {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        Rectangle r = new Rectangle(5, 10);
        System.out.println("Area of circle: " + c.area());
        System.out.println("Perimeter of circle: " +
c.perimeter());
        c.draw_shape(5, 5);

        System.out.println();
        System.out.println("Area of rectangle: " + r.area());
        System.out.println("Perimeter of rectangle: " +
r.perimeter());
        r.draw_shape(5, 10);
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac InterfaceMain.java && java InterfaceMain
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Interf
aceMain.java && java InterfaceMain
Area of circle: 78.53981633974483
Perimeter of circle: 31.41592653589793
Drawing Circle with radius: 5.0

Area of rectangle: 50.0
Perimeter of rectangle: 30.0
Drawing Rectangle with length: 5.0 and breadth: 10.0
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

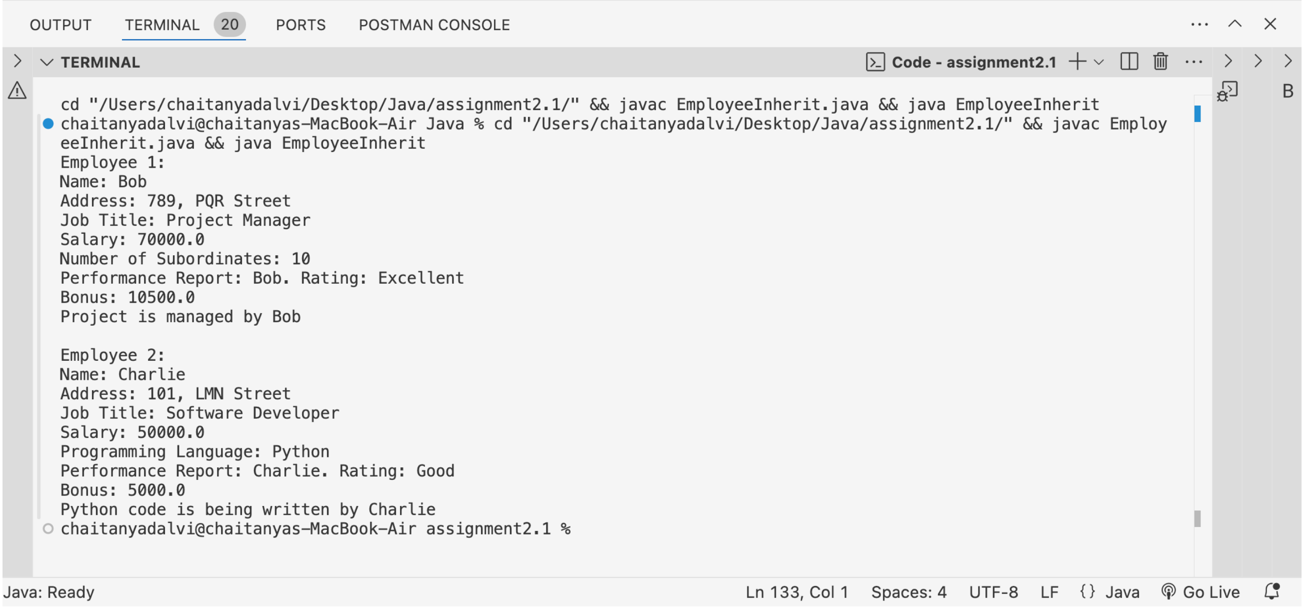
Roll Number: 19

Experiment No: 18

Title: Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager and Developer.

```
class Employee {
    private String name, address, job_title;
    private double salary;

    Employee() {
        name = "John Doe";
        address = "123, XYZ Street";
        job_title = "Software Developer";
        salary = 50000;
    }

    Employee(String name, String address, String job_title, double
salary) {
        this.name = name;
        this.address = address;
        this.job_title = job_title;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public String getJobTitle() {
        return job_title;
    }

    public double getSalary() {
        return salary;
    }

    public String generatePerformanceReport() {
        return ("No performance report available");
    }

    public double calculateBonus() {
        return 0.0;
    }
}

class Manager extends Employee {
    private int numberOfSubordinates;

    Manager() {
        super();
        numberOfSubordinates = 5;
    }

    Manager(String name, String address, String job_title, double
salary, int numberOfSubordinates) {
        super(name, address, job_title, salary);
        this.numberOfSubordinates = numberOfSubordinates;
    }

    public int getNumberOfSubordinates() {
        return numberOfSubordinates;
    }

    public double calculateBonus() {
        return 0.15 * getSalary();
    }

    public void manageProject() {
        System.out.println("Project is managed by " + getName());
    }

    public String generatePerformanceReport() {
        return (getName() + ". Rating: Excellent");
    }
}

class Developer extends Employee {
    private String programmingLanguage;
    Developer() {
        super();
        programmingLanguage = "Java";
    }

    Developer(String name, String address, String job_title, double
salary, String programmingLanguage) {
        super(name, address, job_title, salary);
        this.programmingLanguage = programmingLanguage;
    }

    public String getProgrammingLanguage() {
        return programmingLanguage;
    }

    public double calculateBonus() {
        return 0.1 * getSalary();
    }

    public void writeCode() {
        System.out.println(getProgrammingLanguage() + " code is
being written by " + getName());
    }

    public String generatePerformanceReport() {
        return (getName() + ". Rating: Good");
    }
}

public class EmployeeInherit {
    public static void main(String[] args) {
        Manager emp1 = new Manager("Bob", "789, PQR Street",
"Project Manager", 70000, 10);
        Developer emp2 = new Developer("Charlie", "101, LMN
Street", "Software Developer", 50000, "Python");

        System.out.println("Employee 1:");
        System.out.println("Name: " + emp1.getName());
        System.out.println("Address: " + emp1.getAddress());
        System.out.println("Job Title: " + emp1.getJobTitle());
        System.out.println("Salary: " + emp1.getSalary());
        System.out.println("Number of Subordinates: " +
emp1.getNumberOfSubordinates());
        System.out.println("Performance Report: " +
emp1.generatePerformanceReport());
        System.out.println("Bonus: " + emp1.calculateBonus());
        emp1.manageProject();
        System.out.println();
        System.out.println("Employee 2:");
        System.out.println("Name: " + emp2.getName());
        System.out.println("Address: " + emp2.getAddress());
        System.out.println("Job Title: " + emp2.getJobTitle());
        System.out.println("Salary: " + emp2.getSalary());
        System.out.println("Programming Language: " +
emp2.getProgrammingLanguage());
        System.out.println("Performance Report: " +
emp2.generatePerformanceReport());
        System.out.println("Bonus: " + emp2.calculateBonus());
        emp2.writeCode();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac EmployeeInherit.java && java EmployeeInherit
chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Employ
eeInherit.java && java EmployeeInherit
Employee 1:
Name: Bob
Address: 789, PQR Street
Job Title: Project Manager
Salary: 70000.0
Number of Subordinates: 10
Performance Report: Bob. Rating: Excellent
Bonus: 10500.0
Project is managed by Bob

Employee 2:
Name: Charlie
Address: 101, LMN Street
Job Title: Software Developer
Salary: 50000.0
Programming Language: Python
Performance Report: Charlie. Rating: Good
Bonus: 5000.0
Python code is being written by Charlie
chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

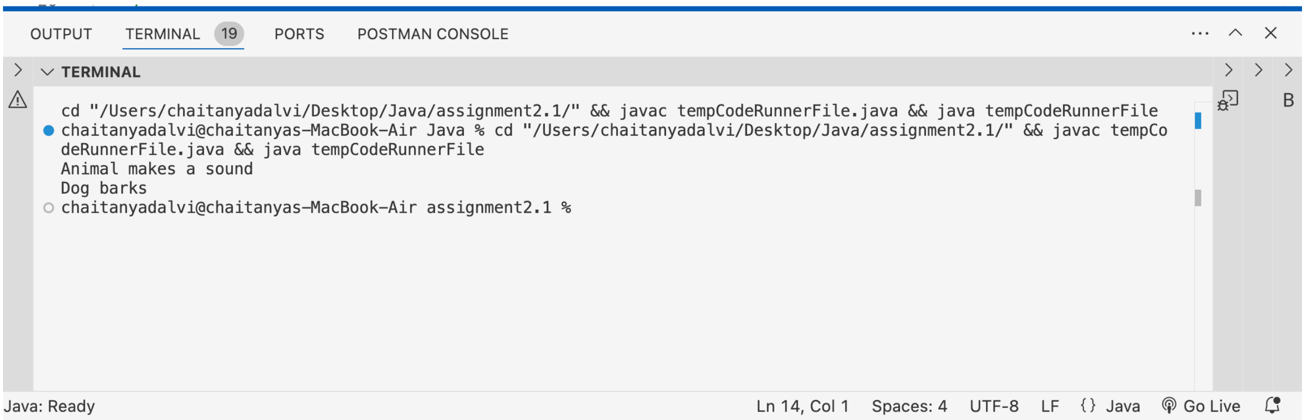
Roll Number: 19

Experiment No: 19

Title: Write a Java program to create an abstract class Animal with an abstract method called sound().

```java
class Animal {
    public void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    public void sound() {
        System.out.println("Dog barks");
    }
}

public class tempCodeRunnerFile {
    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        myAnimal.sound(); // Calls the method from Animal class

        Dog myDog = new Dog();
        myDog.sound(); // Calls the overridden method from Dog
class
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac tempCodeRunnerFile.java && java tempCodeRunnerFile
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac tempCo
deRunnerFile.java && java tempCodeRunnerFile
Animal makes a sound
Dog barks
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

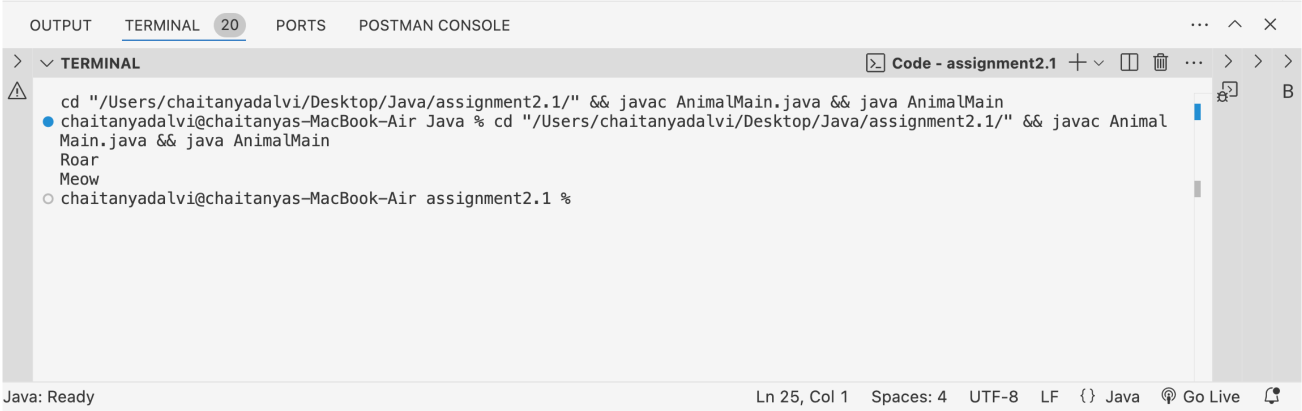
Roll Number: 19

Experiment No: 20

Title: Create subclasses Lion and Cat that extend the Animal class and implement the sound() method to make a specific sound for each animal.

```
abstract class Animal {
    abstract void sound();
}

class Lion extends Animal {
    void sound() {
        System.out.println("Roar");
    }
}

class Cat extends Animal {
    void sound() {
        System.out.println("Meow");
    }
}

public class AnimalMain {
    public static void main(String[] args) {
        Animal lion = new Lion();
        lion.sound();
        Animal cat = new Cat();
        cat.sound();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac AnimalMain.java && java AnimalMain
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Animal
Main.java && java AnimalMain
Roar
Meow
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

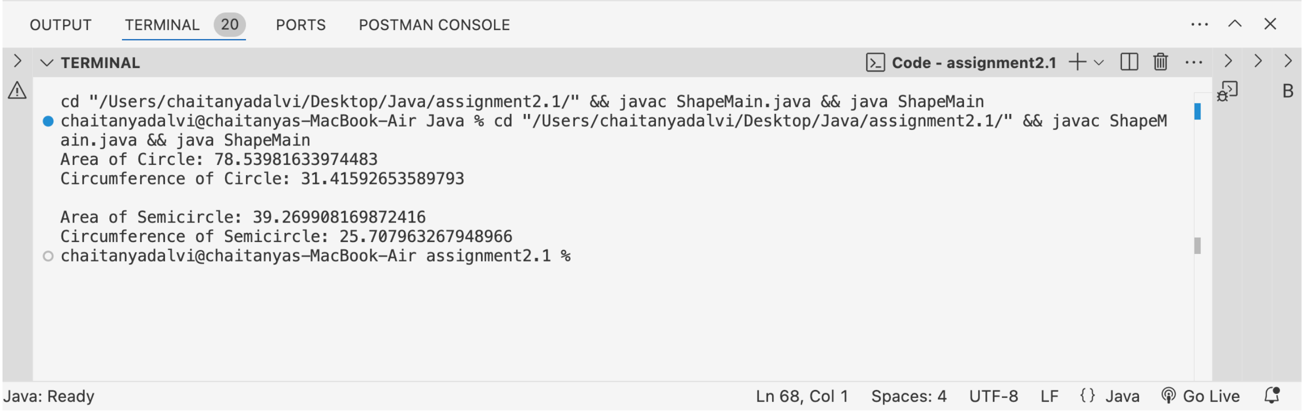
Roll Number: 19

Experiment No: 21

Title: Create an **Abstract Class "Shape"** and 2 subclasses, **SemiCircle and Circle** that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.

```java
abstract class Shape {
    abstract double calculateArea();

    abstract double calculateCircumference();
}

class Circle extends Shape {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    double calculateArea() {
        return Math.PI * radius * radius;
    }

    double calculateCircumference() {
        return 2 * Math.PI * radius;
    }
}

class Semicircle extends Shape {
    double radius;

    Semicircle(double radius) {
        this.radius = radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    double calculateArea() {
        return 0.5 * Math.PI * radius * radius;
    }

    double calculateCircumference() {
        return Math.PI * radius + 2 * radius;
    }
}

public class ShapeMain {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        System.out.println("Area of Circle: " + c.calculateArea());
        System.out.println("Circumference of Circle: " +
c.calculateCircumference() + "\n");

        Semicircle s = new Semicircle(5);
        System.out.println("Area of Semicircle: " +
s.calculateArea());
        System.out.println("Circumference of Semicircle: " +
s.calculateCircumference());
    }
}
```

Output: (screenshot)



```
 cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac ShapeMain.java && java ShapeMain
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac ShapeM
ain.java && java ShapeMain
Area of Circle: 78.53981633974483
Circumference of Circle: 31.41592653589793

Area of Semicircle: 39.269908169872416
Circumference of Semicircle: 25.707963267948966
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

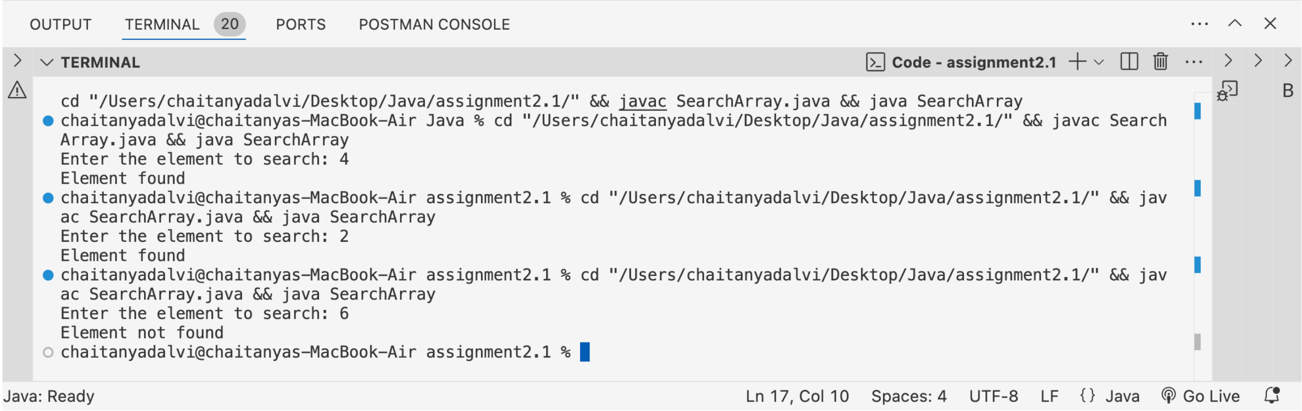
Roll Number: 19

Experiment No: 22

Title: Write a program to search an element in an array using for each loop.

```java
import java.util.Scanner;

public class SearchArray {
    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4, 5 };
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the element to search: ");
        int search = sc.nextInt();
        boolean found = false;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == search) {
                found = true;
                break;
            }
        }
        if (found) {
            System.out.println("Element found");
        } else {
            System.out.println("Element not found");
        }
        sc.close();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SearchArray.java && java SearchArray
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac Search
Array.java && java SearchArray
Enter the element to search: 4
Element found
● chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && jav
ac SearchArray.java && java SearchArray
Enter the element to search: 2
Element found
● chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && jav
ac SearchArray.java && java SearchArray
Enter the element to search: 6
Element not found
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

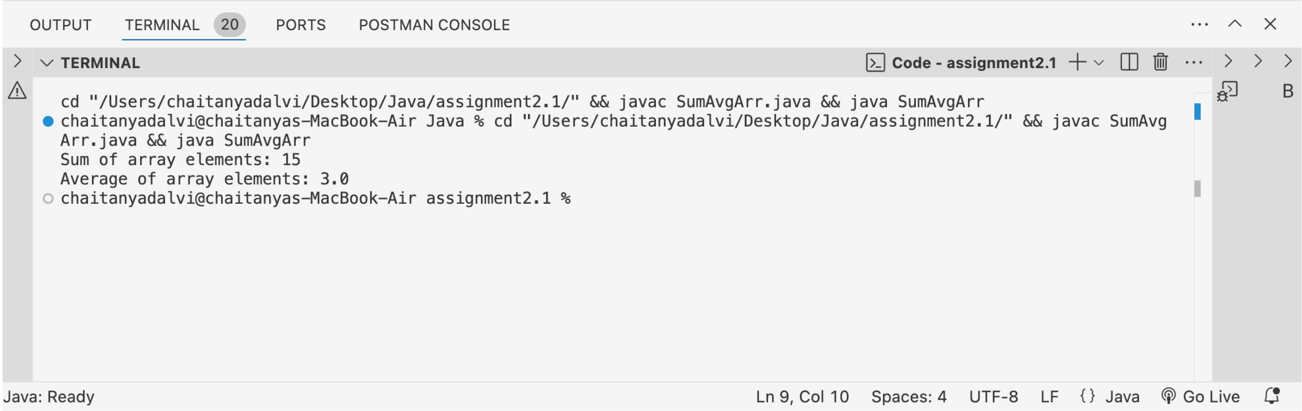
Roll Number: 19

Experiment No: 23

Title: WAP to calculate sum and average of elements stored in one D array.

```java
public class SumAvgArr {
    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4, 5 };
        int sum = 0;
        for (int i = 0; i < arr.length; i++) {
            sum += arr[i];
        }
        System.out.println("Sum of array elements: " + sum);
        System.out.println("Average of array elements: " + (double)
sum / arr.length);
    }

}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SumAvgArr.java && java SumAvgArr
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SumAvg
Arr.java && java SumAvgArr
Sum of array elements: 15
Average of array elements: 3.0
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

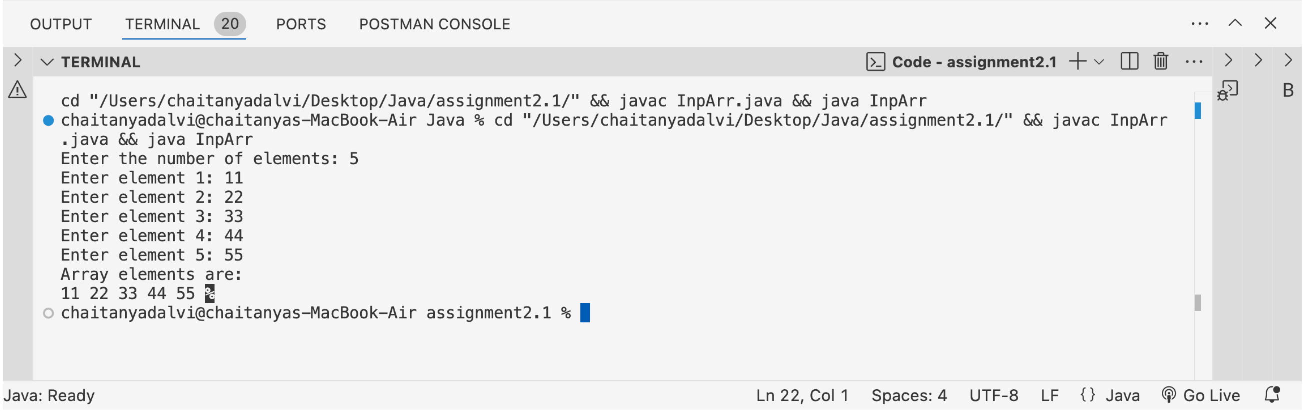
Roll Number: 19

Experiment No: 24

Title: WAP to input elements in one D array using scanner class.

```java
import java.util.Scanner;

public class InpArr {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            arr[i] = sc.nextInt();
        }
        System.out.println("Array elements are:");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        sc.close();
    }
}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac InpArr.java && java InpArr
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac InpArr
.java && java InpArr
Enter the number of elements: 5
Enter element 1: 11
Enter element 2: 22
Enter element 3: 33
Enter element 4: 44
Enter element 5: 55
Array elements are:
11 22 33 44 55 %
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

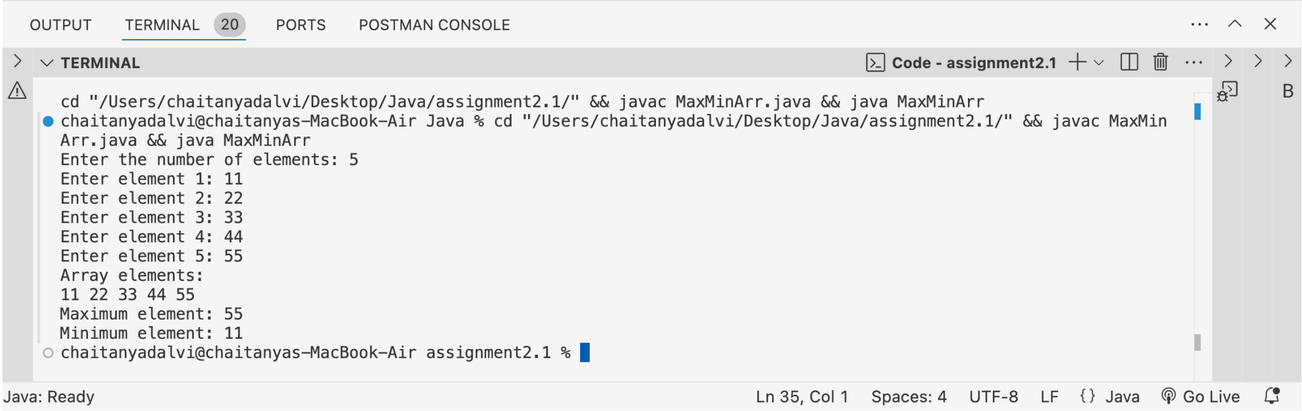
Roll Number: 19

Experiment No: 25

Title: Write a Java program to find the maximum and minimum value of an array.

```java
import java.util.Scanner;

public class MaxMinArr {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            arr[i] = sc.nextInt();
        }
        System.out.println("Array elements: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        int max = arr[0];
        int min = arr[0];
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] > max) {
                max = arr[i];
            }
            if (arr[i] < min) {
                min = arr[i];
            }
        }
        System.out.println("\nMaximum element: " + max);
        System.out.println("Minimum element: " + min);
        sc.close();
    }

}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MaxMinArr.java && java MaxMinArr
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac MaxMin
Arr.java && java MaxMinArr
Enter the number of elements: 5
Enter element 1: 11
Enter element 2: 22
Enter element 3: 33
Enter element 4: 44
Enter element 5: 55
Array elements:
11 22 33 44 55
Maximum element: 55
Minimum element: 11
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```

Name of Student: Chaitanya Dalvi

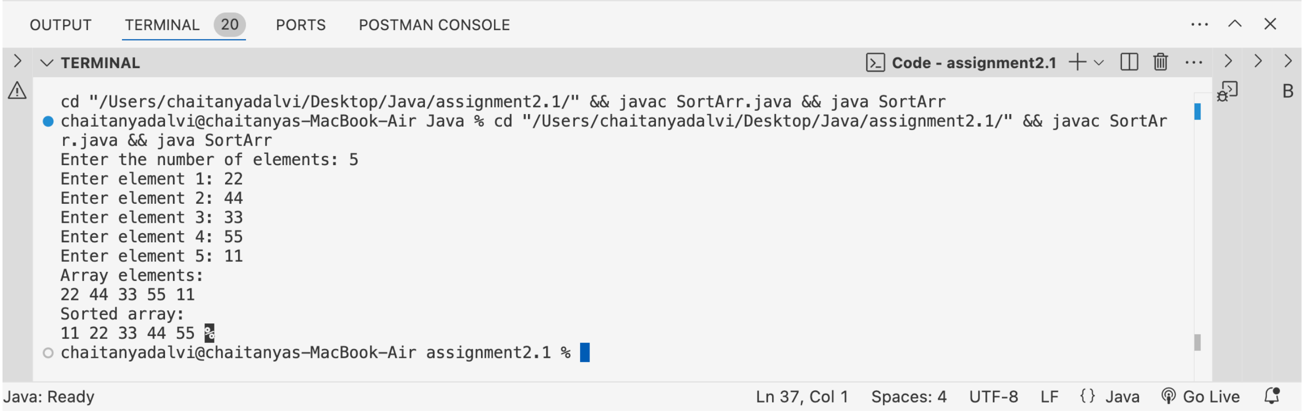
Roll Number: 19

Experiment No: 26

Title: Implement Circular Linked List ADT.

```java
import java.util.Scanner;

public class SortArr {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            arr[i] = sc.nextInt();
        }
        System.out.println("Array elements: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        // bubble sort
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        System.out.println("\nSorted array: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        sc.close();
    }

}
```

Output: (screenshot)



```
cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SortArr.java && java SortArr
● chaitanyadalvi@chaitanyas-MacBook-Air Java % cd "/Users/chaitanyadalvi/Desktop/Java/assignment2.1/" && javac SortAr
r.java && java SortArr
Enter the number of elements: 5
Enter element 1: 22
Enter element 2: 44
Enter element 3: 33
Enter element 4: 55
Enter element 5: 11
Array elements:
22 44 33 55 11
Sorted array:
11 22 33 44 55 %
○ chaitanyadalvi@chaitanyas-MacBook-Air assignment2.1 %
```