

Write the output of the following program.

```
class Test {
    public static void main(String[] args) {
        try {
            System.out.println("Welcome to Java");

            int i = 0;

            int y = 2 / i;

            System.out.println("Welcome to Java");
        }
        catch (ArithmeticException ex) {
            System.out.println("Welcome to Java");
        }
        finally {
            System.out.println("End of the block");
        }
    }
}
```

Welcome to Java

Welcome to Java

End of the block

Write the output of the following program.

```
public abstract class Vehicle {
    protected String brand;
    protected double price;

    public Vehicle() {
        brand = "Unknown";        price    = 50.0;
        System.out.println(" ... Brand : " + brand + " --- Price : " + price); }
    public Vehicle(String b, double p) {
        brand = b;                price    = p;
        System.out.println(" ... Brand : " + brand + " --- Price : " + price); }
    public void show() {
        System.out.println(" ... Brand : " + brand + " --- Price : " + price); }
}

public class Bus extends Vehicle {
    private String name;
    private int nbOfSeats;

    public Bus(){
        name = "Hafeela";
        nbOfSeats = 11;
        show();
    }
    public Bus(String s, String b, double p, int n) {
        name = s;                brand = b;
        price = p;                nbOfSeats = n;
        show();
    }
    public void show() {
        System.out.println(" **** Name : " + name + " .... Nb of Seats : " + nbOfSeats);
        super.show();
    }
    public void addPassangers(int nb) throws Exception{
        if (nb <= 0) throw new Exception ("Unaccepted parameter value");
        if (nb > nbOfSeats) throw new Exception ("Parameter value exceeds available seats");

        nbOfSeats -= nb ;
        show();
    }
}

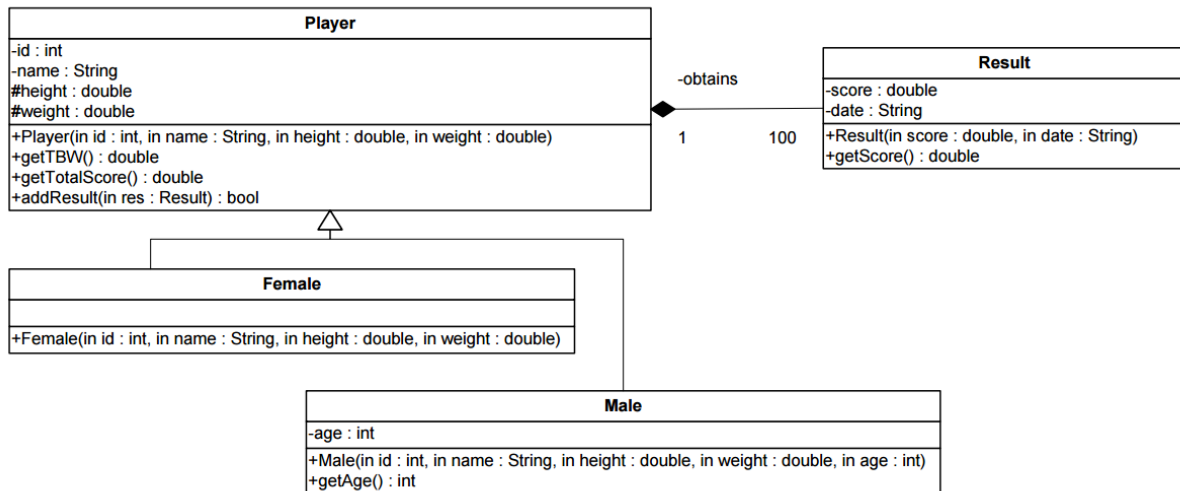
public class Testing {
    public static void main(String[] args) {
        Bus m1 = new Bus();
        System.out.println("++++++");

        Bus m2 = new Bus("m2", "Mercedes", 70.0, 5);
        System.out.println("=====");
        try {
            m1.addPassangers(10);
            System.out.println("-----");
            m2.addPassangers(10);
        }
        catch(Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```
.... Brand : Unknown --- Price : 50.0
**** Name : Hafeela .... Nb of Seats : 11
.... Brand : Unknown --- Price : 50.0
+++++
.... Brand : Unknown --- Price : 50.0
**** Name : m2 .... Nb of Seats : 5
.... Brand : Mercedes --- Price : 70.0
```

```
=====
**** Name : Hafeela .... Nb of Seats : 1
.... Brand : Unknown --- Price : 50.0
-----
```

Parameter value exceeds available seats



- **Class Player**
 - **Player(...):** constructor. By default a player may have 100 results.
 - **getTBW():** calculates the Total Body Water (TBW) based on the following formulas:
 - For **Male**: $TBW = 2.447 - (0.09156 \times \text{age}) + (0.1074 \times \text{height}) + (0.3362 \times \text{weight})$
 - For **Female**: $TBW = -2.097 + (0.1069 \times \text{height}) + (0.2466 \times \text{weight})$
 - **getTotalScore():** returns the sum of the scores obtained by the player.
 - **addResult(...):** adds a new result to the player results. It returns true if the insertion is done. Otherwise, it returns false.
- **Class Result**
 - **Result(...):** constructor
 - **getScore():** returns the score of the result. This method throws an exception “No Score” if Score equals zero.
- **Class Male**
 - **Male(...):** constructor
 - **getAge():** returns the age of the male.

Write in Java the classes: Player, Result and Male.

```

public class Result {

    private double score;
    private String date;

    public Result(double s, String d) {
        this.score = s;
        this.date = d;
    }

    public Result(Result r) {
        this.score = r.score;
        this.date = r.date;
    }

    public double getScore() throws Exception {
        if(score == 0)
            throw new Exception("No Score");
        return score;
    }
}

public abstract class Player {
    private int id;
    private String name;
    protected double height;
    protected double weight;
    private Result[] obtained;
    int nb;

    public Player(int id, String name, double height, double weight) {
        this.id = id;
        this.name = name;
        this.height = height;
        this.weight = weight;
        obtained = new Result[100];
        nb = 0;
    }

    public abstract double getTBW();

    public double getTotalScore() {
        double sum = 0.0;
        for (int i = 0; i < nb; i++) {
            try {
                sum += obtained[i].getScore();
            } catch (Exception e) {
                System.out.print(e.getMessage());
            }
        }
        return sum;
    }
}

```

```
    public boolean addResult(Result res) {
        if (nb < obtained.length) {
            obtained[nb] = new Result(res);
            nb++;
            return true;
        }
        return false;
    }
}
```

```
public class Male extends Player {
    private int age;

    public Male(int id, String name, double height, double weight, int age) {
        super(id, name, height, weight);
        this.age = age;
    }

    public int getAge() {
        return age;
    }

    public double getTBW() {
        return 2.447 - (0.09156 * age) + (0.1074 * height) + (0.3362 * weight);
    }
}
```