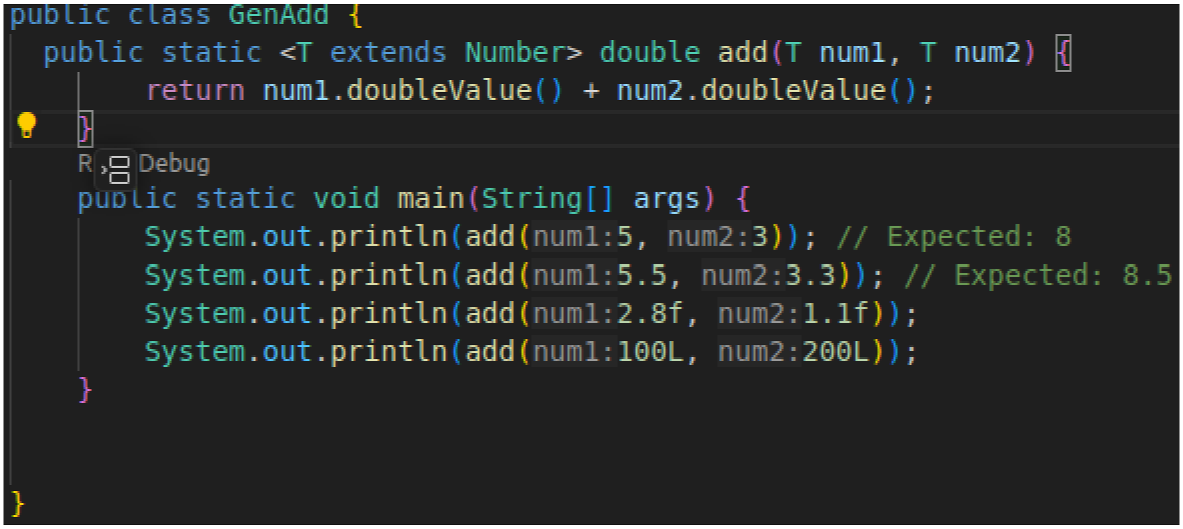
PG-DAC THIRUVANANTHAPURAM & KOCHI
OOPs WITH JAVA

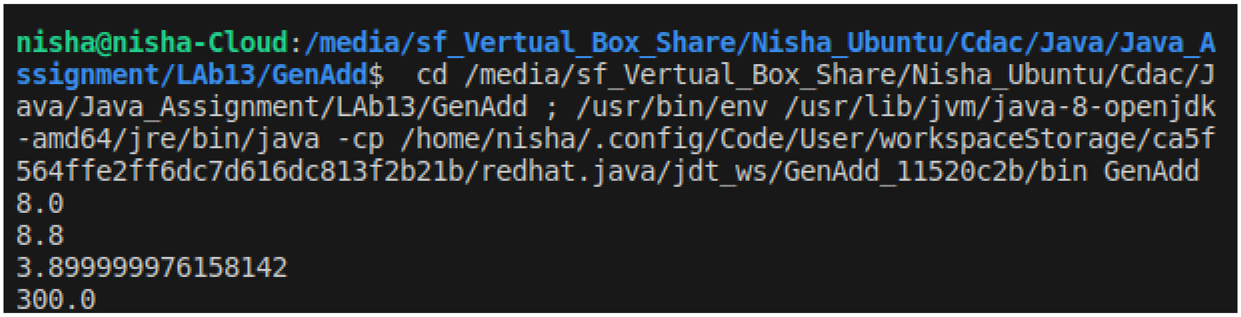
Generic Methods/Classes/Interfaces

Q1.

Write a Java program to create a generic method that takes two arguments of the same type adds them and return.



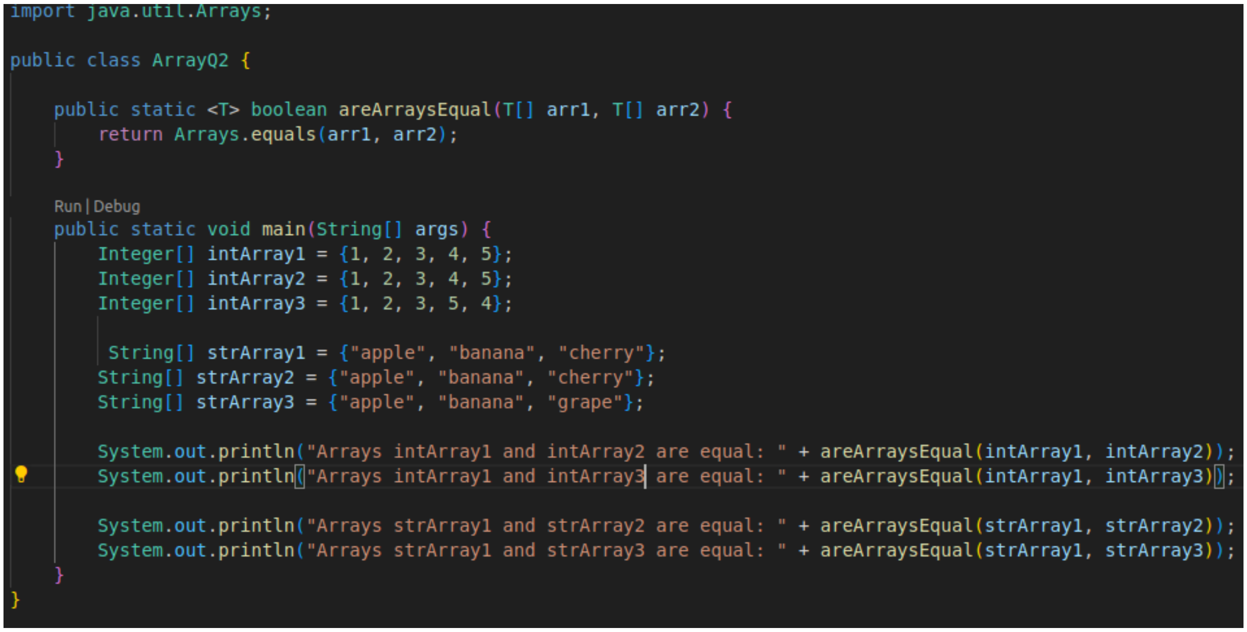
```
public class GenAdd {
  public static <T extends Number> double add(T num1, T num2) {
        return num1.doubleValue() + num2.doubleValue();
    }
    public static void main(String[] args) {
        System.out.println(add(num1:5, num2:3)); // Expected: 8
        System.out.println(add(num1:5.5, num2:3.3)); // Expected: 8.5
        System.out.println(add(num1:2.8f, num2:1.1f));
        System.out.println(add(num1:100L, num2:200L));
    }


}
```



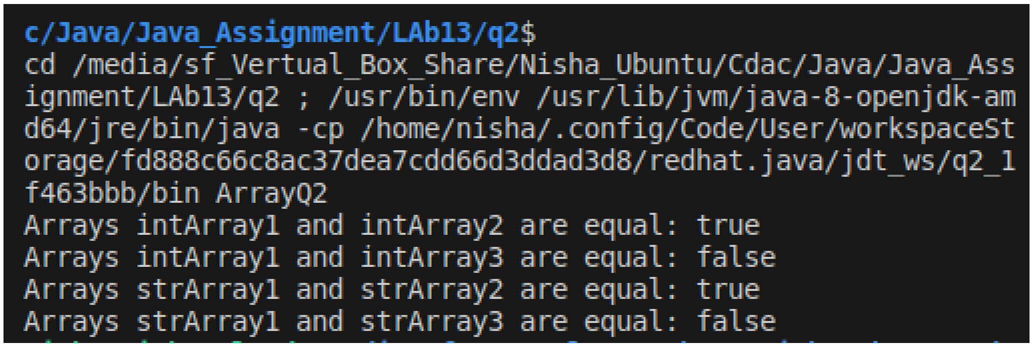
```
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_A
ssignment/LAb13/GenAdd$  cd /media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/J
ava/Java_Assignment/LAb13/GenAdd ; /usr/bin/env /usr/lib/jvm/java-8-openjdk
-amd64/jre/bin/java -cp /home/nisha/.config/Code/User/workspaceStorage/ca5f
564ffe2ff6dc7d616dc813f2b21b/redhat.java/jdt_ws/GenAdd_11520c2b/bin GenAdd
8.0
8.8
3.899999976158142
300.0
```

Q2.

Write a Java program to create a generic method that takes two arrays of the same type and checks if they have the same elements in the same order.



```
import java.util.Arrays;

public class ArrayQ2 {

    public static <T> boolean areArraysEqual(T[] arr1, T[] arr2) {
        return Arrays.equals(arr1, arr2);
    }

    Run | Debug
    public static void main(String[] args) {
        Integer[] intArray1 = {1, 2, 3, 4, 5};
        Integer[] intArray2 = {1, 2, 3, 4, 5};
        Integer[] intArray3 = {1, 2, 3, 5, 4};

         String[] strArray1 = {"apple", "banana", "cherry"};
        String[] strArray2 = {"apple", "banana", "cherry"};
        String[] strArray3 = {"apple", "banana", "grape"};

        System.out.println("Arrays intArray1 and intArray2 are equal: " + areArraysEqual(intArray1, intArray2));
        System.out.println("Arrays intArray1 and intArray3 are equal: " + areArraysEqual(intArray1, intArray3));

        System.out.println("Arrays strArray1 and strArray2 are equal: " + areArraysEqual(strArray1, strArray2));
        System.out.println("Arrays strArray1 and strArray3 are equal: " + areArraysEqual(strArray1, strArray3));
    }
}
```



```
c/Java/Java_Assignment/LAb13/q2$
cd /media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Ass
ignment/LAb13/q2 ; /usr/bin/env /usr/lib/jvm/java-8-openjdk-am
d64/jre/bin/java -cp /home/nisha/.config/Code/User/workspaceSt
orage/fd888c66c8ac37dea7cdd66d3ddad3d8/redhat.java/jdt_ws/q2_1
f463bbb/bin ArrayQ2
Arrays intArray1 and intArray2 are equal: true
Arrays intArray1 and intArray3 are equal: false
Arrays strArray1 and strArray2 are equal: true
Arrays strArray1 and strArray3 are equal: false
```

Q3.

Create a Generic Class Weather with an normal/no-generic data member named day and a generic data member temperature.

Create

a) a constructor to initialize the data members and

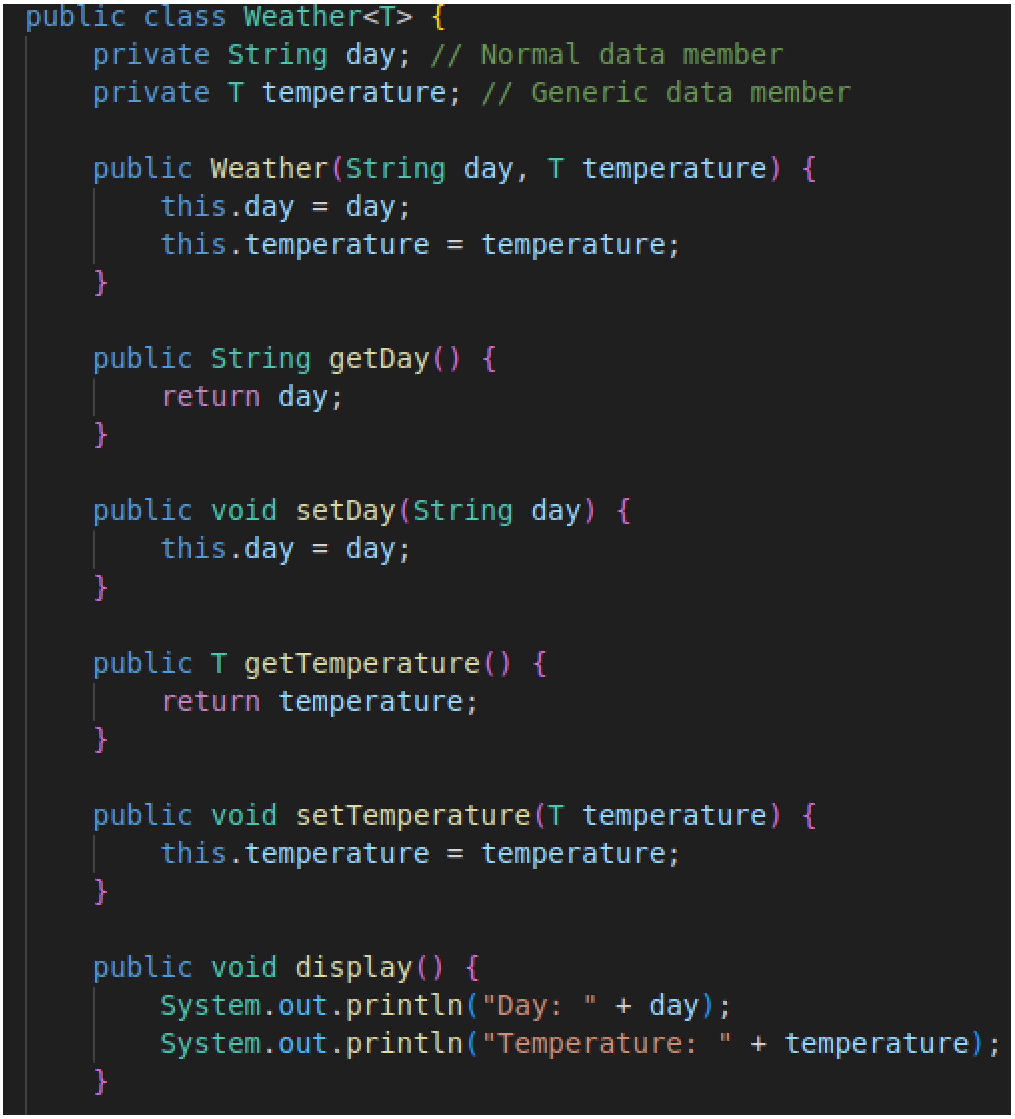
b) getters and setters for both the data members

c) a normal/no-generic method display() to display the data

i) Create instance of the class from main

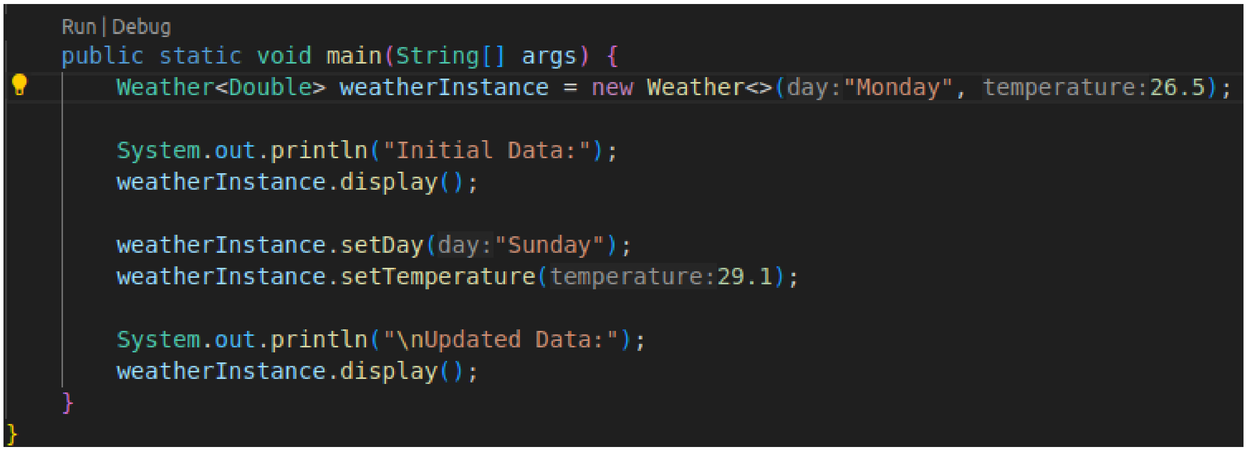
ii) call getters and setters for retrieving/ changing the data.

iii) display the data with the display() method

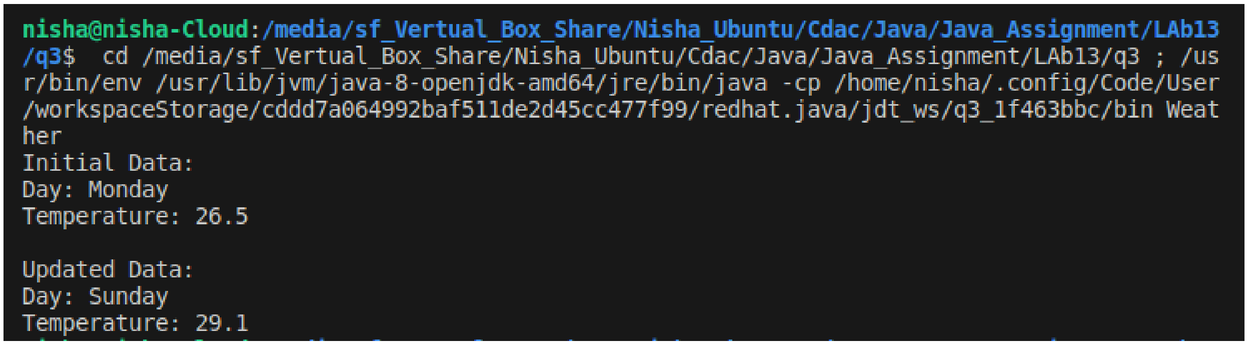


```
public class Weather<T> {
    private String day; // Normal data member
    private T temperature; // Generic data member

    public Weather(String day, T temperature) {
        this.day = day;
        this.temperature = temperature;
    }

    public String getDay() {
        return day;
    }

    public void setDay(String day) {
        this.day = day;
    }

    public T getTemperature() {
        return temperature;
    }

    public void setTemperature(T temperature) {
        this.temperature = temperature;
    }

    public void display() {
        System.out.println("Day: " + day);
        System.out.println("Temperature: " + temperature);
    }
```

.



```
    Run | Debug
    public static void main(String[] args) {
        Weather<Double> weatherInstance = new Weather<>(day:"Monday", temperature:26.5);

        System.out.println("Initial Data:");
        weatherInstance.display();

        weatherInstance.setDay(day:"Sunday");
        weatherInstance.setTemperature(temperature:29.1);

        System.out.println("\nUpdated Data:");
        weatherInstance.display();
    }
}
```

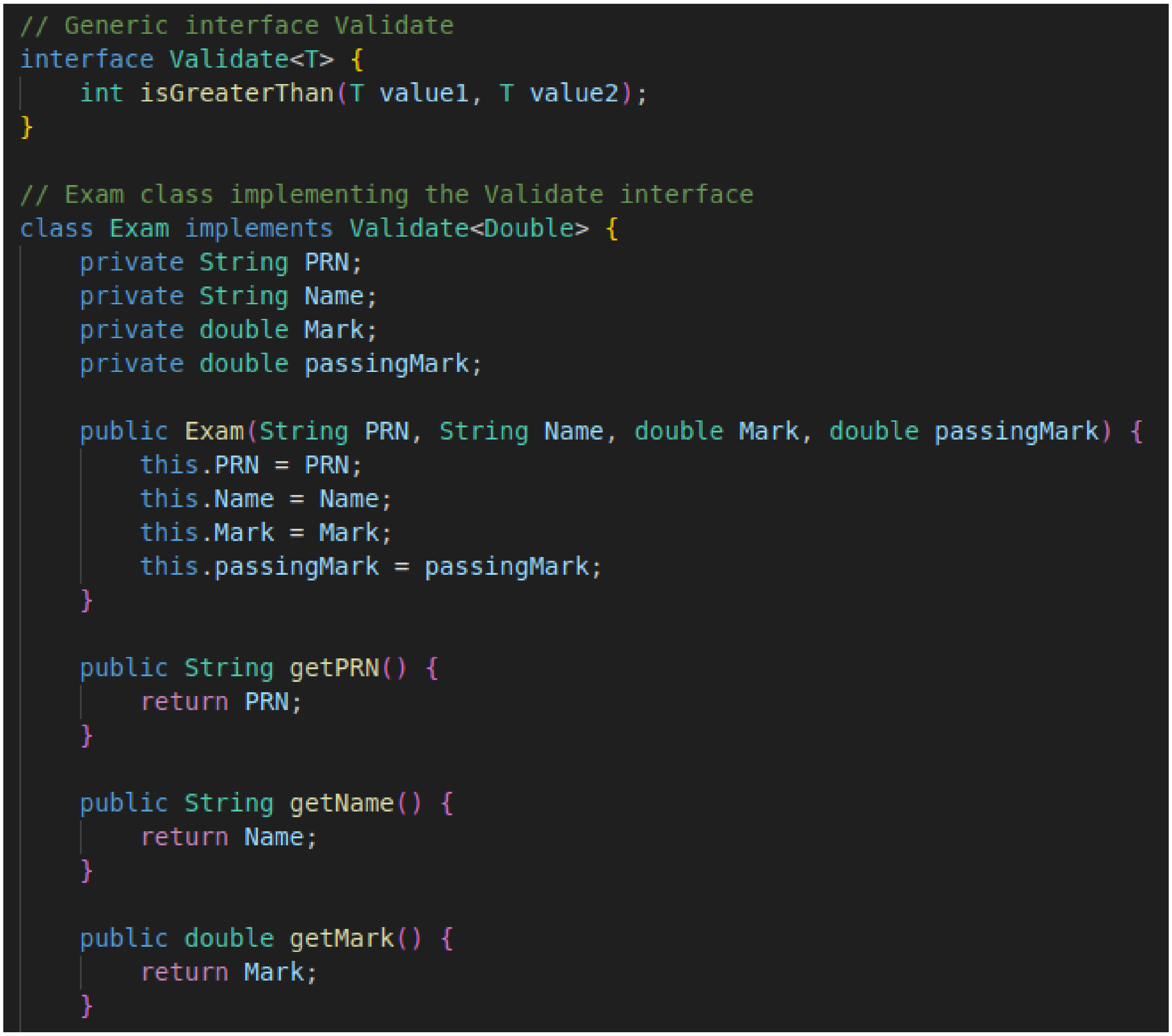


```
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Assignment/LAb13
/q3$  cd /media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Assignment/LAb13/q3 ; /us
r/bin/env /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -cp /home/nisha/.config/Code/User
/workspaceStorage/cddd7a064992baf511de2d45cc477f99/redhat.java/jdt_ws/q3_1f463bbc/bin Weat
her
Initial Data:
Day: Monday
Temperature: 26.5

Updated Data:
Day: Sunday
Temperature: 29.1
```

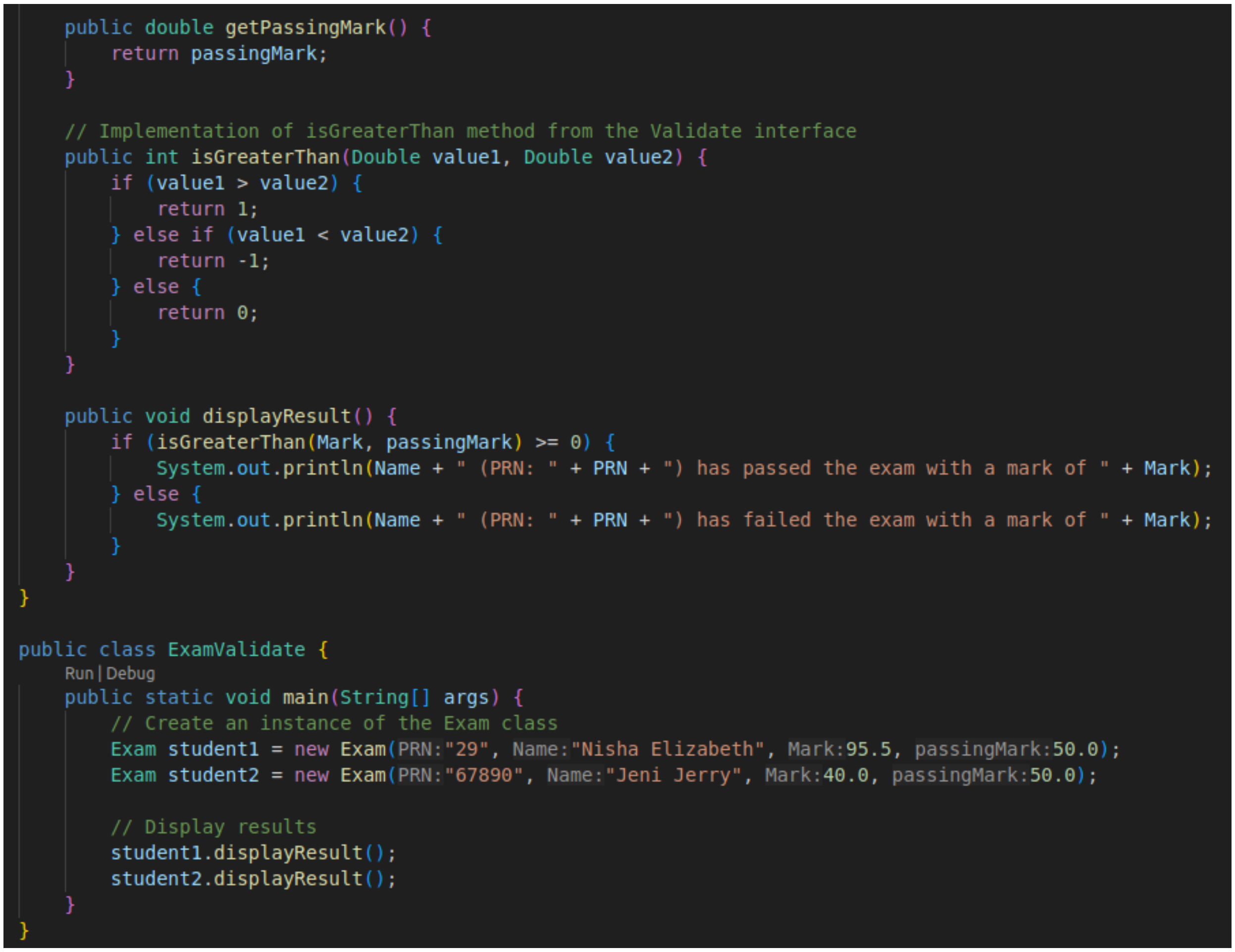
Q4.

Create a Generic Interface Validate with a generic method isGreaterThan() that takes 2 generic parameters of the same type and return an integer.

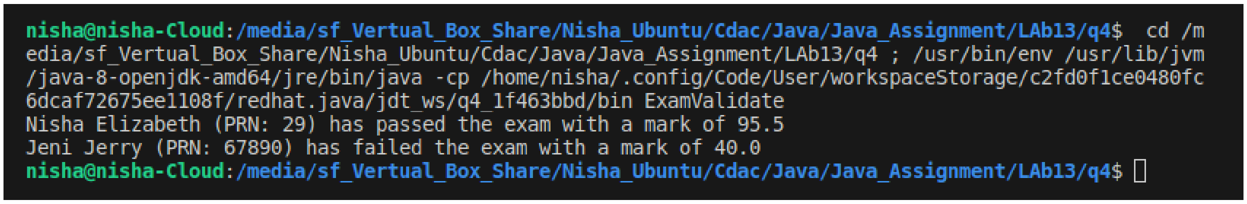
Create an Exam class with PRN, Name & Mark as data memebers. Exam class must implement the Validate interface to check if Mark of student is greater than the minimum mark for passing.



```java
// Generic interface Validate
interface Validate<T> {
    int isGreaterThan(T value1, T value2);
}

// Exam class implementing the Validate interface
class Exam implements Validate<Double> {
    private String PRN;
    private String Name;
    private double Mark;
    private double passingMark;

    public Exam(String PRN, String Name, double Mark, double passingMark) {
        this.PRN = PRN;
        this.Name = Name;
        this.Mark = Mark;
        this.passingMark = passingMark;
    }

    public String getPRN() {
        return PRN;
    }

    public String getName() {
        return Name;
    }

    public double getMark() {
        return Mark;
    }
```



```
    public double getPassingMark() {
        return passingMark;
    }

    // Implementation of isGreaterThan method from the Validate interface
    public int isGreaterThan(Double value1, Double value2) {
        if (value1 > value2) {
            return 1;
        } else if (value1 < value2) {
            return -1;
        } else {
            return 0;
        }
    }

    public void displayResult() {
        if (isGreaterThan(Mark, passingMark) >= 0) {
            System.out.println(Name + " (PRN: " + PRN + ") has passed the exam with a mark of " + Mark);
        } else {
            System.out.println(Name + " (PRN: " + PRN + ") has failed the exam with a mark of " + Mark);
        }
    }
}

public class ExamValidate {
    Run | Debug
    public static void main(String[] args) {
        // Create an instance of the Exam class
        Exam student1 = new Exam(PRN:"29", Name:"Nisha Elizabeth", Mark:95.5, passingMark:50.0);
        Exam student2 = new Exam(PRN:"67890", Name:"Jeni Jerry", Mark:40.0, passingMark:50.0);

        // Display results
        student1.displayResult();
        student2.displayResult();
    }
}
```



```
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Assignment/LAb13/q4$  cd /m
edia/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Assignment/LAb13/q4 ; /usr/bin/env /usr/lib/jvm
/java-8-openjdk-amd64/jre/bin/java -cp /home/nisha/.config/Code/User/workspaceStorage/c2fd0f1ce0480fc
6dcaf72675ee1108f/redhat.java/jdt_ws/q4_1f463bbd/bin ExamValidate
Nisha Elizabeth (PRN: 29) has passed the exam with a mark of 95.5
Jeni Jerry (PRN: 67890) has failed the exam with a mark of 40.0
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/Java/Java_Assignment/LAb13/q4$ 
```