Student’s ID: AF0402433 **Trainer’s Name: Manali Ma’am**
Student’s Name: Patel Abubakar Siddique Mehboob

LAB 6

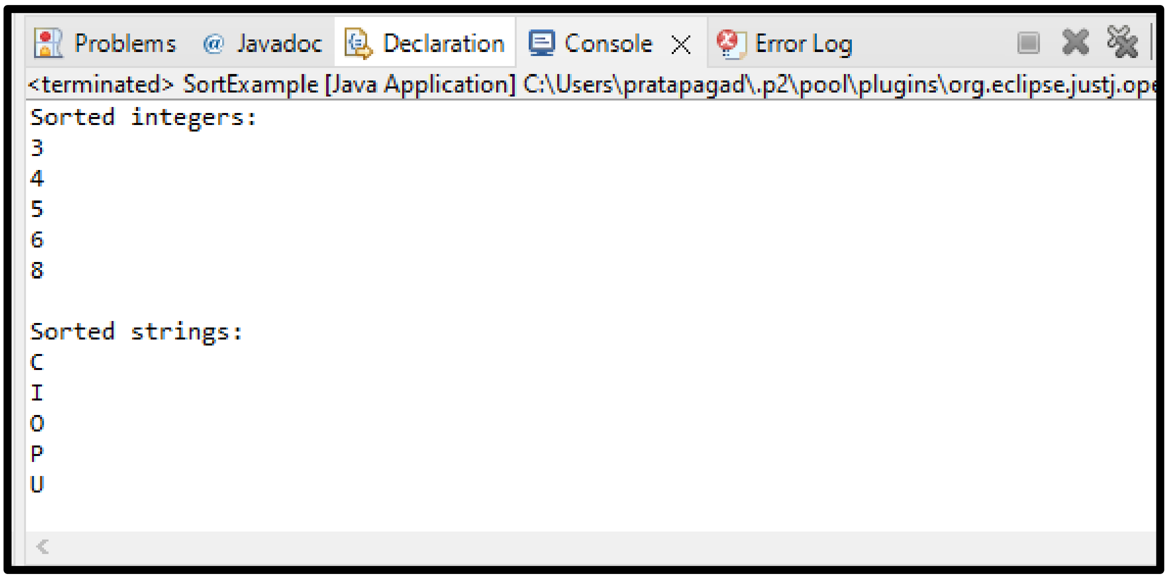
Q.1. Write a java programme to sort the integers 8, 4, 3, 5, 6 and the alphabetical string C, O, I, P, U, in ascending order. Show the resulting output.

Program:

```
package Lab6;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class SortExample {
    public static void main(String[] args) {
        // Integer arraylist
        List<Integer> numbers = new ArrayList<>();
        numbers.add(8);
        numbers.add(4);
        numbers.add(3);
        numbers.add(5);
        numbers.add(6);

        // String arraylist
        List<String> strings = new ArrayList<>();
        strings.add("C");
        strings.add("O");
        strings.add("I");
        strings.add("P");
        strings.add("U");

        // Sorting integers
        Collections.sort(numbers);

        // Sorting strings
        Collections.sort(strings);

        // Printing sorted integers
        System.out.println("Sorted integers:");//printing the result
        for (Integer num : numbers) {
            System.out.println(num);
        }

        // Printing sorted strings
        System.out.println("\nSorted strings:");//printing the result
        for (String str : strings) {
            System.out.println(str);
        }
    }
}
```

Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Output:



```
<terminated> SortExample [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclipse.justj.ope
Sorted integers:
3
4
5
6
8

Sorted strings:
C
I
O
P
U
```

Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

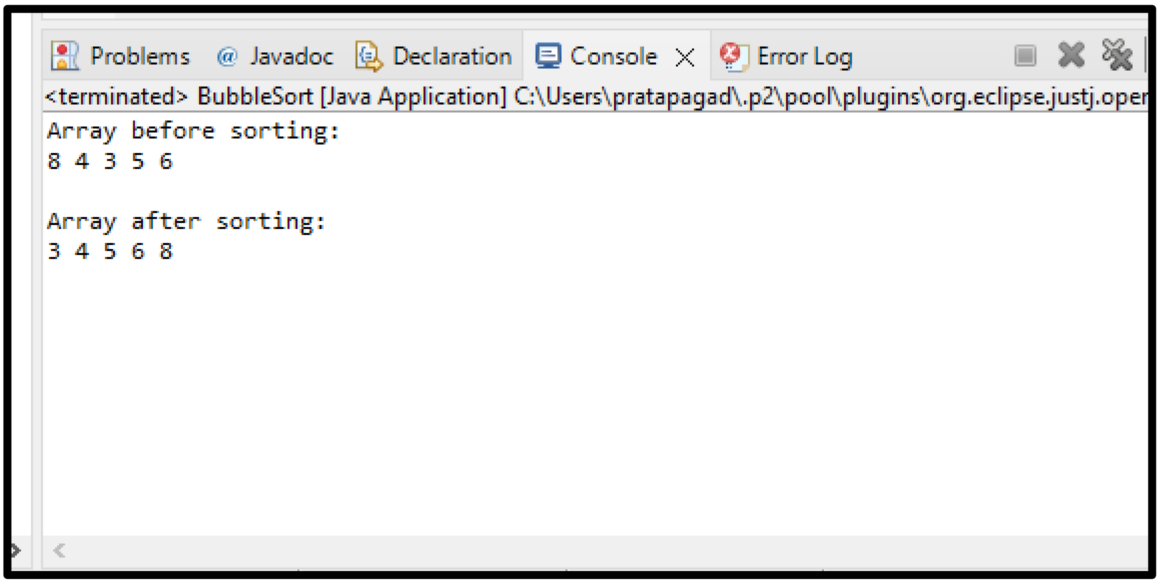
Q.2. Write a Java program to implement the bubble sort algorithm to sort an array of integers in ascending order.

Program:

```
package Lab6;
public class BubbleSort {
    public static void main(String[] args) {
        int[] array = {8, 4, 3, 5, 6};

        System.out.println("Array before sorting:");
        printArray(array);

        bubbleSort(array);

        System.out.println("\nArray after sorting:");
        printArray(array);
    }

    // Bubble sort algorithm implementation
    public static void bubbleSort(int[] array) {
        int n = array.length;
        boolean swapped;

        for (int i = 0; i < n - 1; i++) {
            swapped = false;
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    // Swap array[j] and array[j + 1]
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                    swapped = true;
                }
            }
            // If no two elements were swapped in the inner loop, then
break
            if (!swapped) {
                break;
            }
        }
    }

    // Utility method to print an array
    public static void printArray(int[] array) {
        for (int num : array) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
}
```

Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Output:



```
<terminated> BubbleSort [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclipse.justj.oper
Array before sorting:
8 4 3 5 6

Array after sorting:
3 4 5 6 8
```

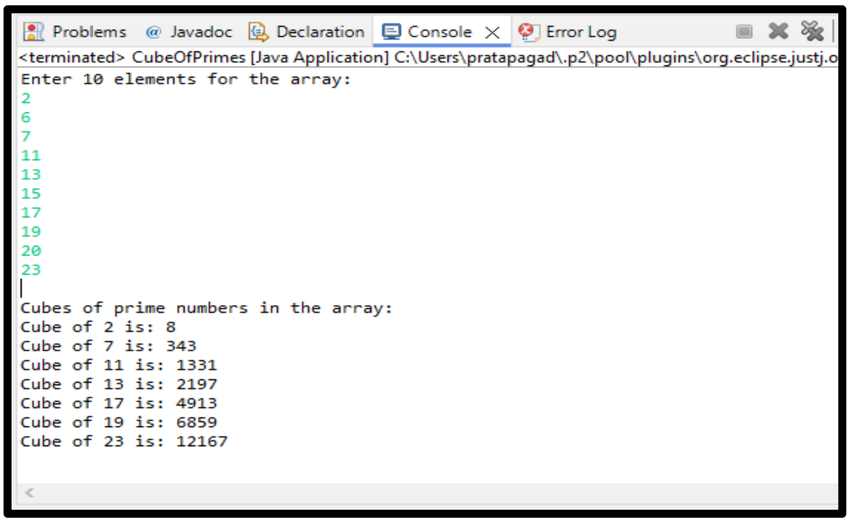
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.3 Write a program to input an array 10 elements and print the cube of prime numbers in it.

Program:

```
package Lab6;
import java.util.Scanner;
public class CubeOfPrimes {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Input array of 10 elements
        int[] array = new int[10];
        System.out.println("Enter 10 elements for the array:");
        for (int i = 0; i < 10; i++) {
            array[i] = scanner.nextInt();
        }
        // Print the cubes of prime numbers in the array
        System.out.println("\nCubes of prime numbers in the array:");
        for (int num : array) {
            if (isPrime(num)) {
                long cube = (long) num * num * num; // Calculate cube
                System.out.println("Cube of " + num + " is: " + cube);
            }
        }
        scanner.close();    }

    // Method to check if a number is prime
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

Output:



```
<terminated> CubeOfPrimes [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclipse.justj.o
Enter 10 elements for the array:
2
6
7
11
13
15
17
19
20
23

Cubes of prime numbers in the array:
Cube of 2 is: 8
Cube of 7 is: 343
Cube of 11 is: 1331
Cube of 13 is: 2197
Cube of 17 is: 4913
Cube of 19 is: 6859
Cube of 23 is: 12167
```

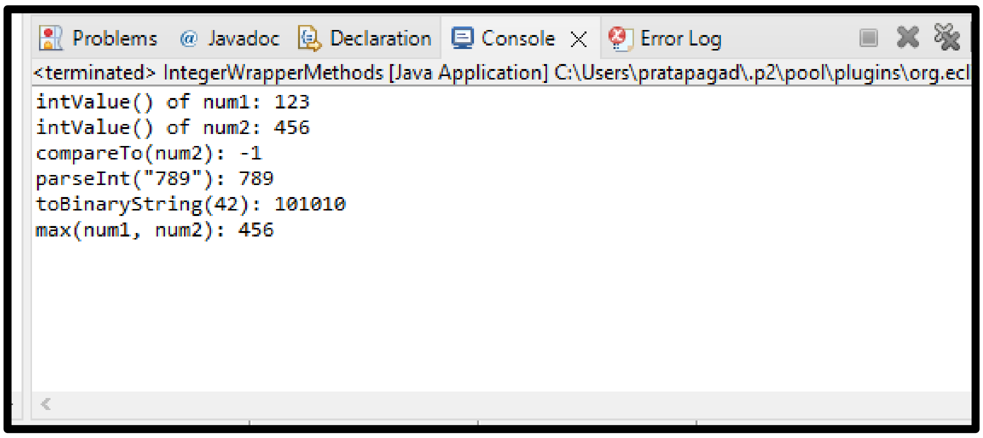
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.4. Write a java program to implement integer wrapper class methods. (Any 5 methods).

Program:

```java
package Lab6;
public class IntegerWrapperMethods {
    public static void main(String[] args) {
        // Creating Integer objects
        Integer num1 = 123;
        Integer num2 = 456;

        // 1. intValue() - Returns the value of this Integer as an int
        int value1 = num1.intValue();
        int value2 = num2.intValue();
        System.out.println("intValue() of num1: " + value1);
        System.out.println("intValue() of num2: " + value2);

        // 2. compareTo(Integer anotherInteger) - Compares two Integer 
objects numerically
        int compareResult = num1.compareTo(num2);
        System.out.println("compareTo(num2): " + compareResult);

        // 3. static parseInt(String s) - Parses the string argument as a 
signed decimal integer
        String str = "789";
        int parsedInt = Integer.parseInt(str);
        System.out.println("parseInt(\"789\"): " + parsedInt);

        // 4. static toBinaryString(int i) - Returns a string 
representation of the integer argument as an unsigned integer in base 2
        int number = 42;
        String binaryString = Integer.toBinaryString(number);
        System.out.println("toBinaryString(42): " + binaryString);

        // 5. static max(int a, int b) - Returns the greater of two int 
values
        int maxNumber = Integer.max(num1, num2);
        System.out.println("max(num1, num2): " + maxNumber);
    }
}
```

Output:



```
<terminated> IntegerWrapperMethods [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.ecl
intValue() of num1: 123
intValue() of num2: 456
compareTo(num2): -1
parseInt("789"): 789
toBinaryString(42): 101010
max(num1, num2): 456
```

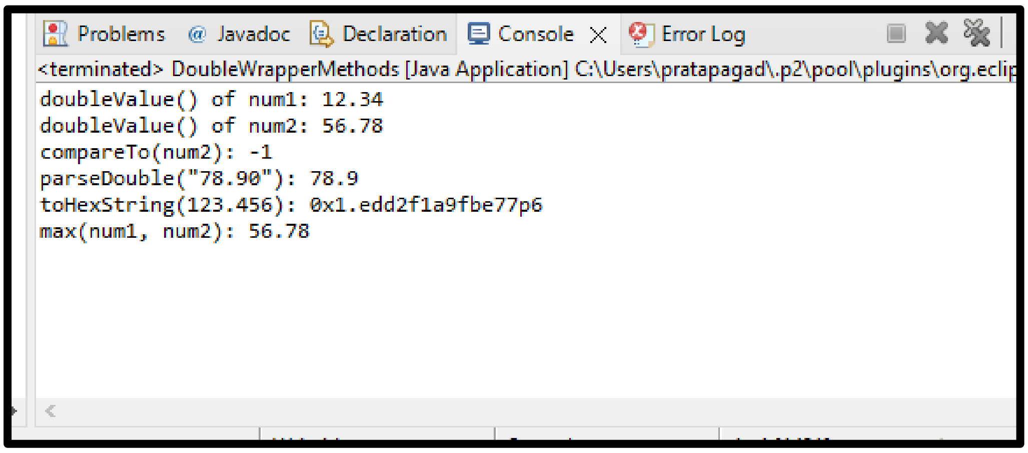
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.5. Write a java program to implement double wrapper class methods. (Any 5 methods).

Program:

```
package Lab6;
public class DoubleWrapperMethods {
    public static void main(String[] args) {
        // Creating Double objects
        Double num1 = 12.34;
        Double num2 = 56.78;

        // 1. doubleValue() - Returns the value of this Double as a double
        double value1 = num1.doubleValue();
        double value2 = num2.doubleValue();
        System.out.println("doubleValue() of num1: " + value1);
        System.out.println("doubleValue() of num2: " + value2);

        // 2. compareTo(Double anotherDouble) - Compares two Double
objects numerically
        int compareResult = num1.compareTo(num2);
        System.out.println("compareTo(num2): " + compareResult);

        // 3. static parseDouble(String s) - Parses the string argument as
a double
        String str = "78.90";
        double parsedDouble = Double.parseDouble(str);
        System.out.println("parseDouble(\"78.90\"): " + parsedDouble);

        // 4. static toHexString(double d) - Returns a hexadecimal string
representation of the double argument
        double number = 123.456;
        String hexString = Double.toHexString(number);
        System.out.println("toHexString(123.456): " + hexString);

        // 5. static max(double a, double b) - Returns the greater of two
double values
        double maxNumber = Double.max(num1, num2);
        System.out.println("max(num1, num2): " + maxNumber);
    }
}
```

Output:



```
<terminated> DoubleWrapperMethods [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclip
doubleValue() of num1: 12.34
doubleValue() of num2: 56.78
compareTo(num2): -1
parseDouble("78.90"): 78.9
toHexString(123.456): 0x1.edd2f1a9fbe77p6
max(num1, num2): 56.78
```

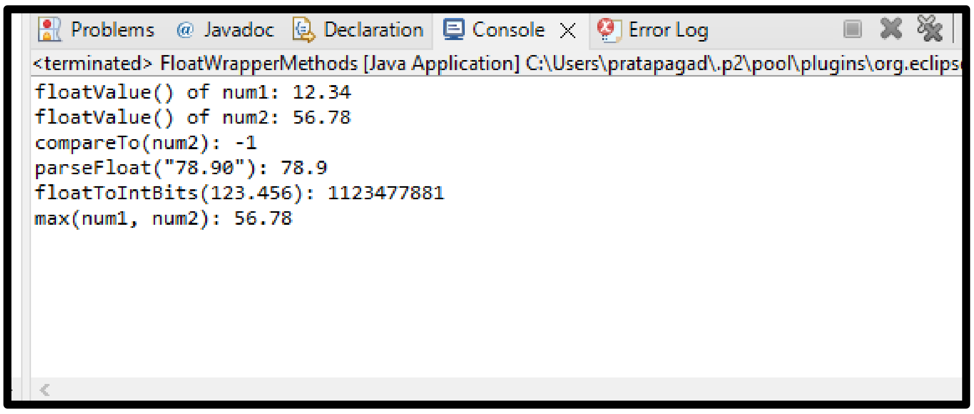
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.6. Write a java program to implement float wrapper class methods. (Any 5 methods).

Program:

```
package Lab6;
public class FloatWrapperMethods {
    public static void main(String[] args) {
        // Creating Float objects
        Float num1 = 12.34f;
        Float num2 = 56.78f;

        // 1. floatValue() - Returns the value of this Float as a float
        float value1 = num1.floatValue();
        float value2 = num2.floatValue();
        System.out.println("floatValue() of num1: " + value1);
        System.out.println("floatValue() of num2: " + value2);

        // 2. compareTo(Float anotherFloat) - Compares two Float objects
numerically
        int compareResult = num1.compareTo(num2);
        System.out.println("compareTo(num2): " + compareResult);

        // 3. static parseFloat(String s) - Parses the string argument as
a float
        String str = "78.90";
        float parsedFloat = Float.parseFloat(str);
        System.out.println("parseFloat(\"78.90\"): " + parsedFloat);

        // 4. static floatToIntBits(float value) - Returns a
representation of the specified floating-point value according to the IEEE
754 floating-point "single format" bit layout
        float number = 123.456f;
        int floatBits = Float.floatToIntBits(number);
        System.out.println("floatToIntBits(123.456): " + floatBits);

        // 5. static max(float a, float b) - Returns the greater of two
float values
        float maxNumber = Float.max(num1, num2);
        System.out.println("max(num1, num2): " + maxNumber);
    }
}
```

Output:



```
<terminated> FloatWrapperMethods [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclips
floatValue() of num1: 12.34
floatValue() of num2: 56.78
compareTo(num2): -1
parseFloat("78.90"): 78.9
floatToIntBits(123.456): 1123477881
max(num1, num2): 56.78
```

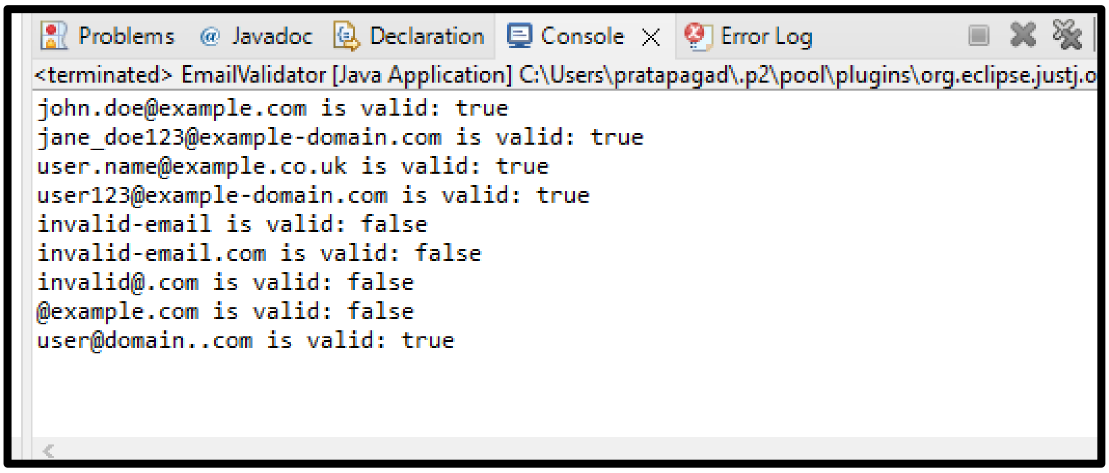
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.7. Write a Java program to validate email addresses using regular expressions. The email should have the format username@domain.com where username and domain can contain alphanumeric characters, dots, and hyphens.

Program:

```java
package Lab6;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class EmailValidator {
    public static void main(String[] args) {
        String[] emails = {
            "john.doe@example.com",
            "jane_doe123@example-domain.com",
            "user.name@example.co.uk",
            "user123@example-domain.com",
            "invalid-email",
            "invalid-email.com",
            "invalid@.com",
            "@example.com",
            "user@domain..com"
        };
        // Regular expression pattern for email validation
        String regex = "^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";
        Pattern pattern = Pattern.compile(regex);
        // Validate each email
        for (String email : emails) {
            boolean isValid = validateEmail(email, pattern);
            System.out.println(email + " is valid: " + isValid);
        }
    }
    // Method to validate email using regex pattern
    public static boolean validateEmail(String email, Pattern pattern) {
        Matcher matcher = pattern.matcher(email);
        return matcher.matches();
    }
}
```

Output:



```
<terminated> EmailValidator [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclipse.justj.o
john.doe@example.com is valid: true
jane_doe123@example-domain.com is valid: true
user.name@example.co.uk is valid: true
user123@example-domain.com is valid: true
invalid-email is valid: false
invalid-email.com is valid: false
invalid@.com is valid: false
@example.com is valid: false
user@domain..com is valid: true
```

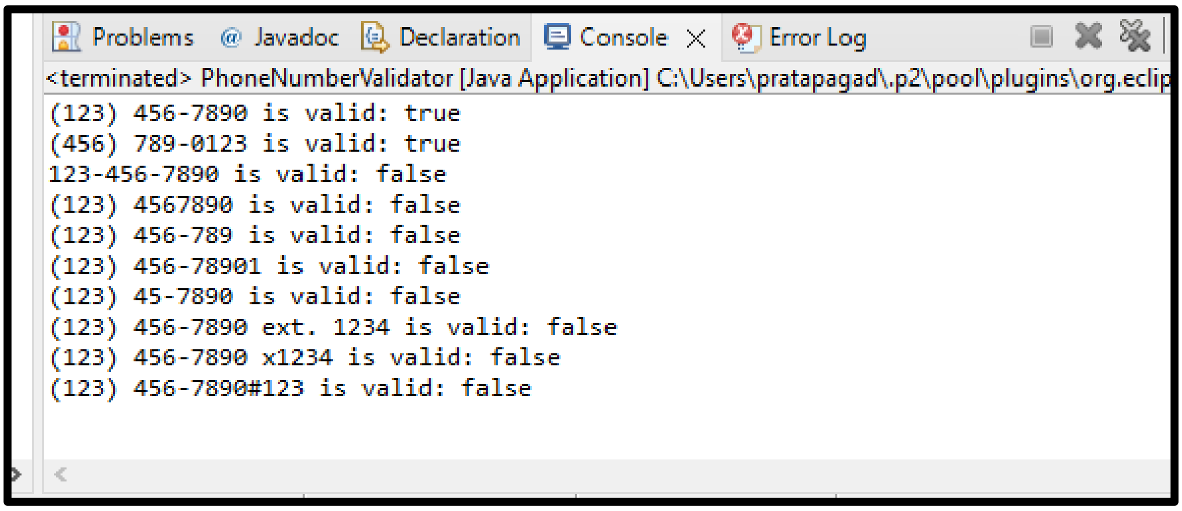
Student's ID: AF0402433 **Trainer's Name: Manali Ma'am**
Student's Name: Patel Abubakar Siddique Mehboob

Q.8. Create a Java program to validate phone numbers. The format should be (xxx) xxx-xxxx where x is a digit.

Program:

```java
package Lab6;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class PhoneNumberValidator {
    public static void main(String[] args) {
        String[] phoneNumbers = {
            "(123) 456-7890",
            "(456) 789-0123",
            "123-456-7890",
            "(123) 4567890",
            "(123) 456-789",
            "(123) 456-78901",
            "(123) 45-7890",
            "(123) 456-7890 ext. 1234",
            "(123) 456-7890 x1234",
            "(123) 456-7890#123"
        };
        // Regular expression pattern for phone number validation
        String regex = "^\\(\\d{3}\\) \\d{3}-\\d{4}$";
        Pattern pattern = Pattern.compile(regex);
        // Validate each phone number
        for (String phoneNumber : phoneNumbers) {
            boolean isValid = validatePhoneNumber(phoneNumber, pattern);
            System.out.println(phoneNumber + " is valid: " + isValid);
        }
    }
    // Method to validate phone number using regex pattern
    public static boolean validatePhoneNumber(String phoneNumber, Pattern 
pattern) {
        Matcher matcher = pattern.matcher(phoneNumber);
        return matcher.matches();
    }
}
```

Output:



```
<terminated> PhoneNumberValidator [Java Application] C:\Users\pratapagad\.p2\pool\plugins\org.eclip
(123) 456-7890 is valid: true
(456) 789-0123 is valid: true
123-456-7890 is valid: false
(123) 4567890 is valid: false
(123) 456-789 is valid: false
(123) 456-78901 is valid: false
(123) 45-7890 is valid: false
(123) 456-7890 ext. 1234 is valid: false
(123) 456-7890 x1234 is valid: false
(123) 456-7890#123 is valid: false
```