Name: 117_Swapnil Dhotre _KH

OOPJ Assignment - 2

1. Working with java.lang.Boolean

a. Explore the Java API documentation for java.lang.Boolean and observe its modifiers and super types.

b. Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method.

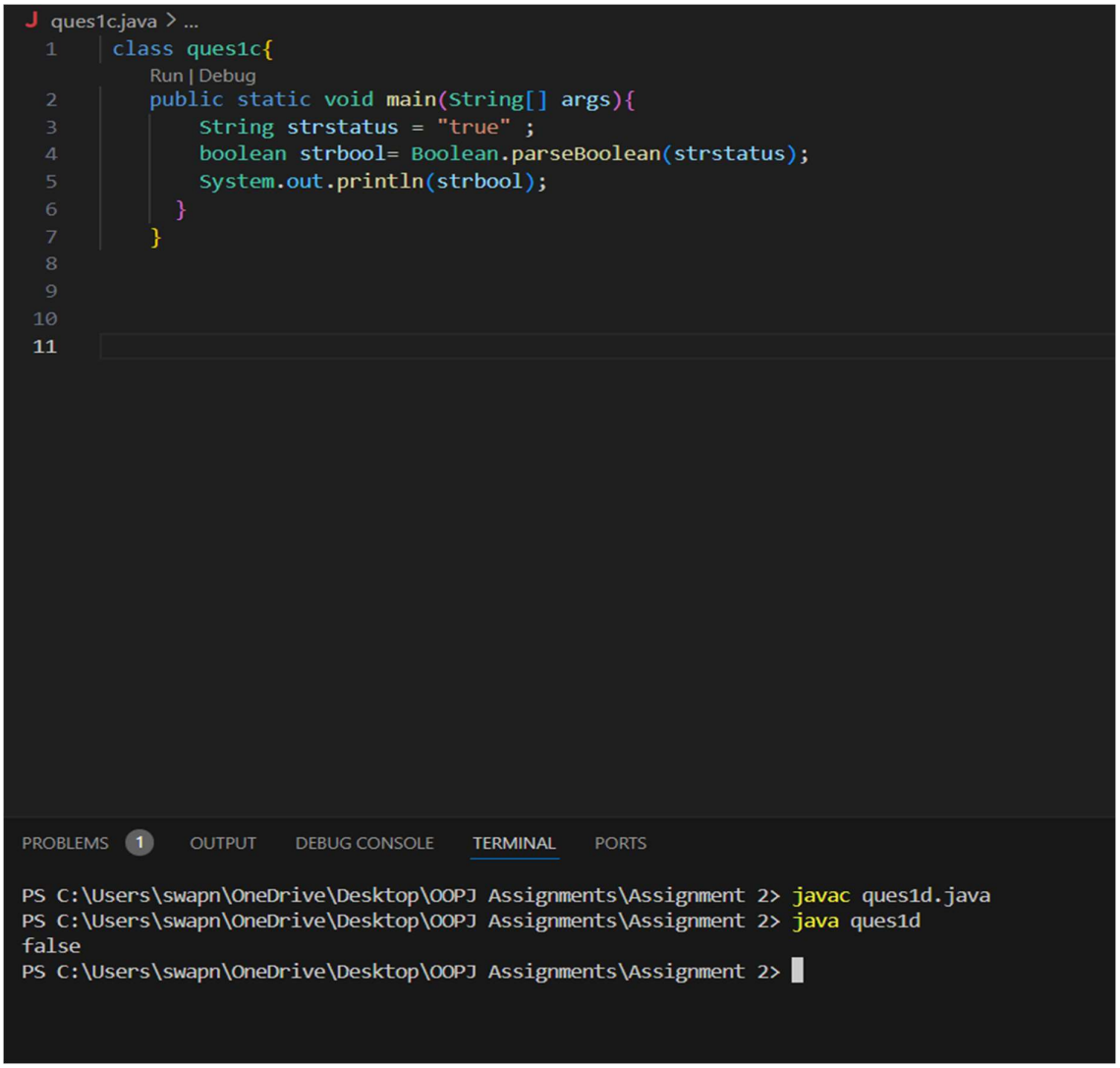
ques1b.java > ...

```java
class ques1b{
Run | Debug
public static void main(String[] args){
    boolean status =true;
    String bool = Boolean.toString(status) ;
    System.out.println(x:"status");
  }
}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

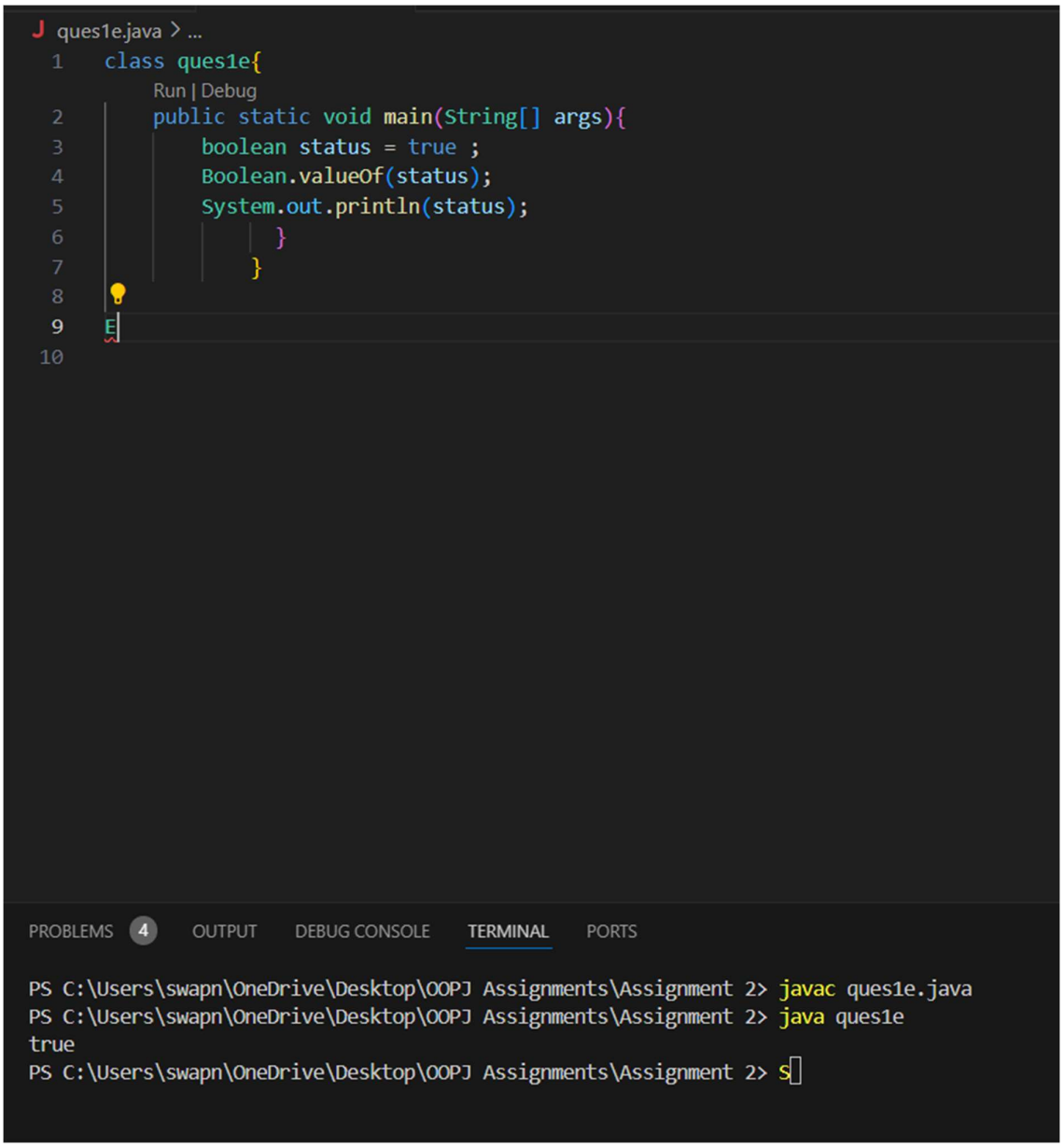
```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1b
status
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

c. Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method.

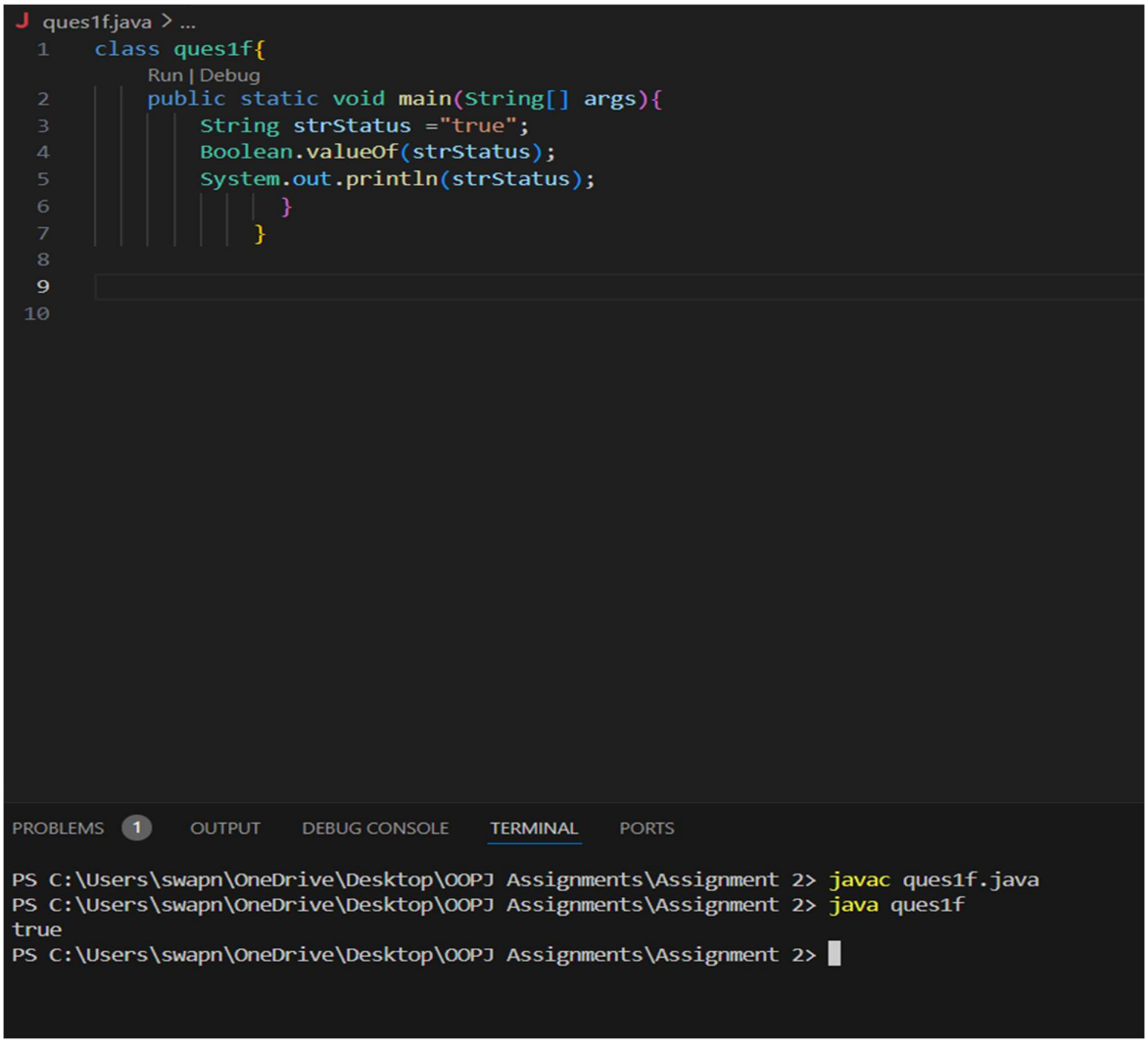


d. Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean.

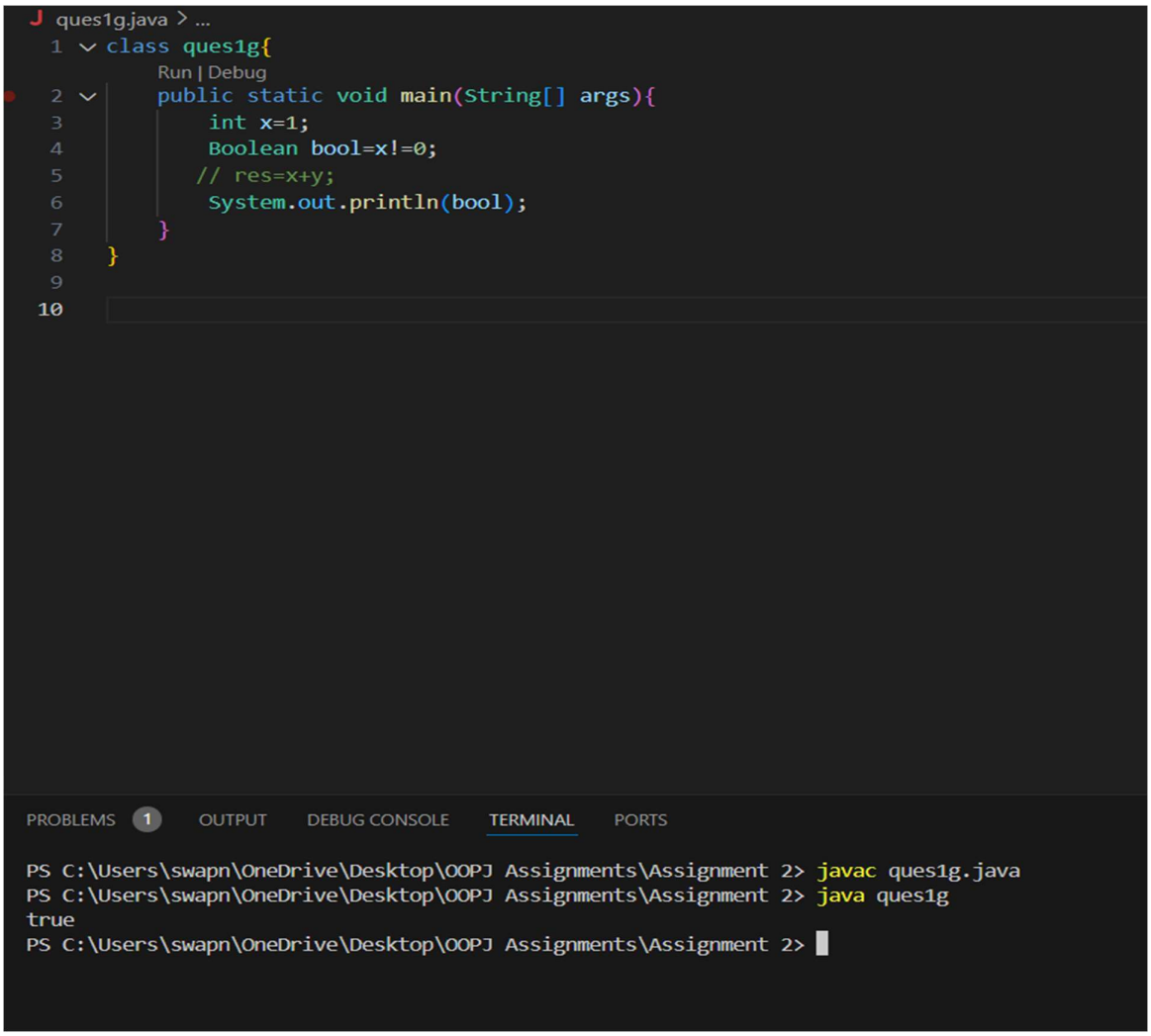
e. Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf().



f. Declare a method-local variable strStatus of type String with the value "true" and convert it to the corresponding wrapper class using Boolean.valueOf().



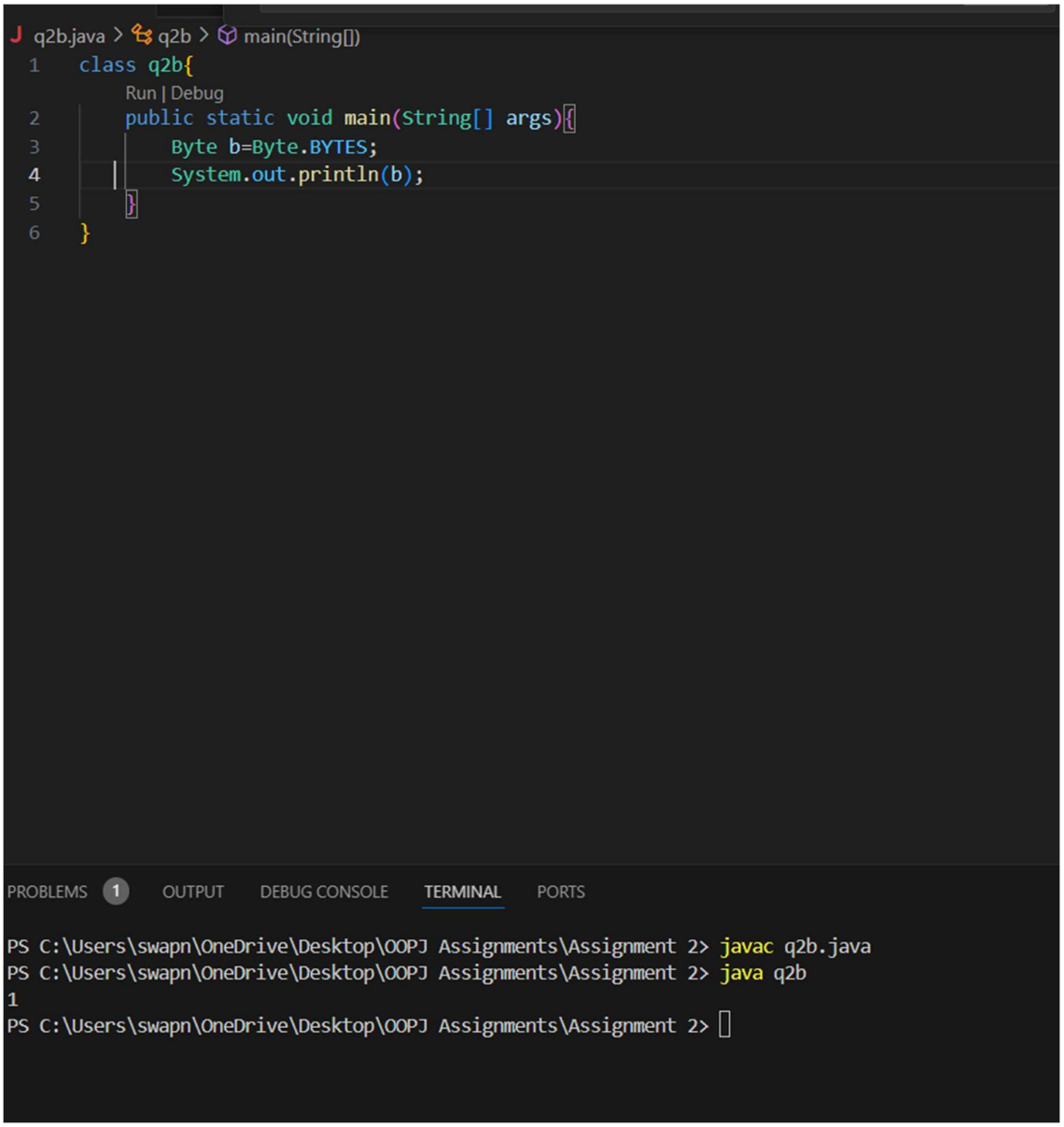
g. Experiment with converting a boolean value into other primitive types or vice versa and observe the results.



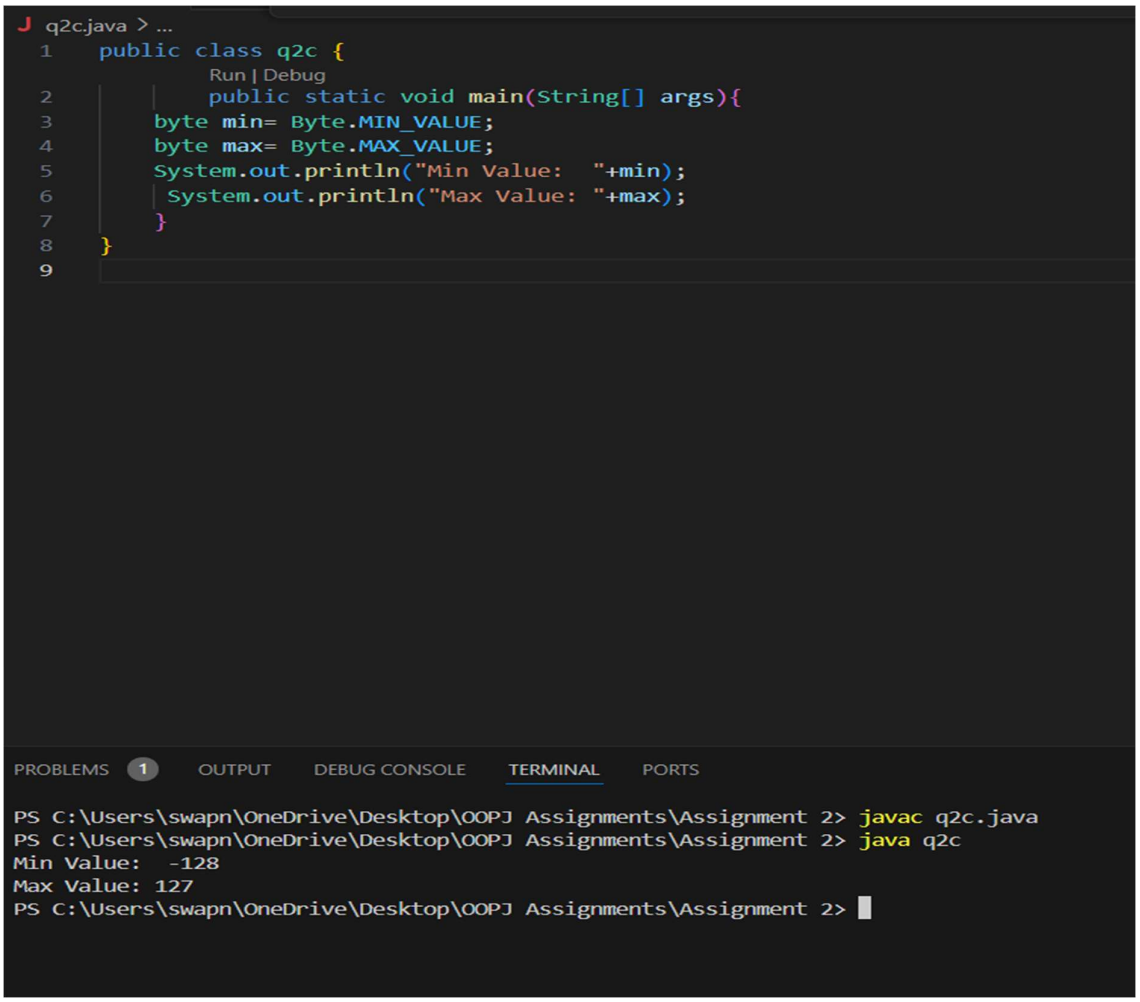
2. Working with java.lang.Byte

a. Explore the Java API documentation for java.lang.Byte and observe its modifiers and super types.

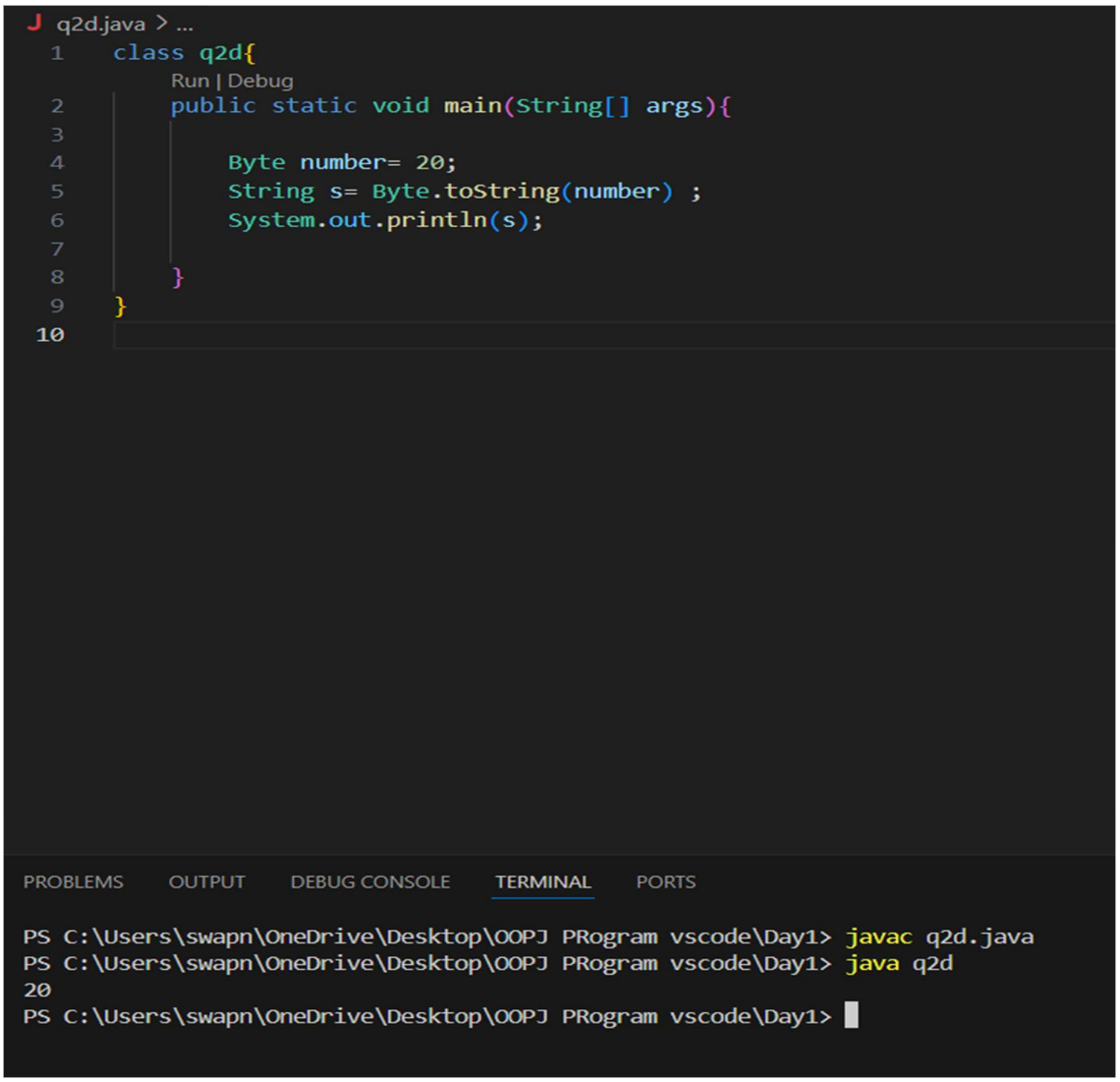
b. Write a program to test how many bytes are used to represent a byte value using the BYTES field.



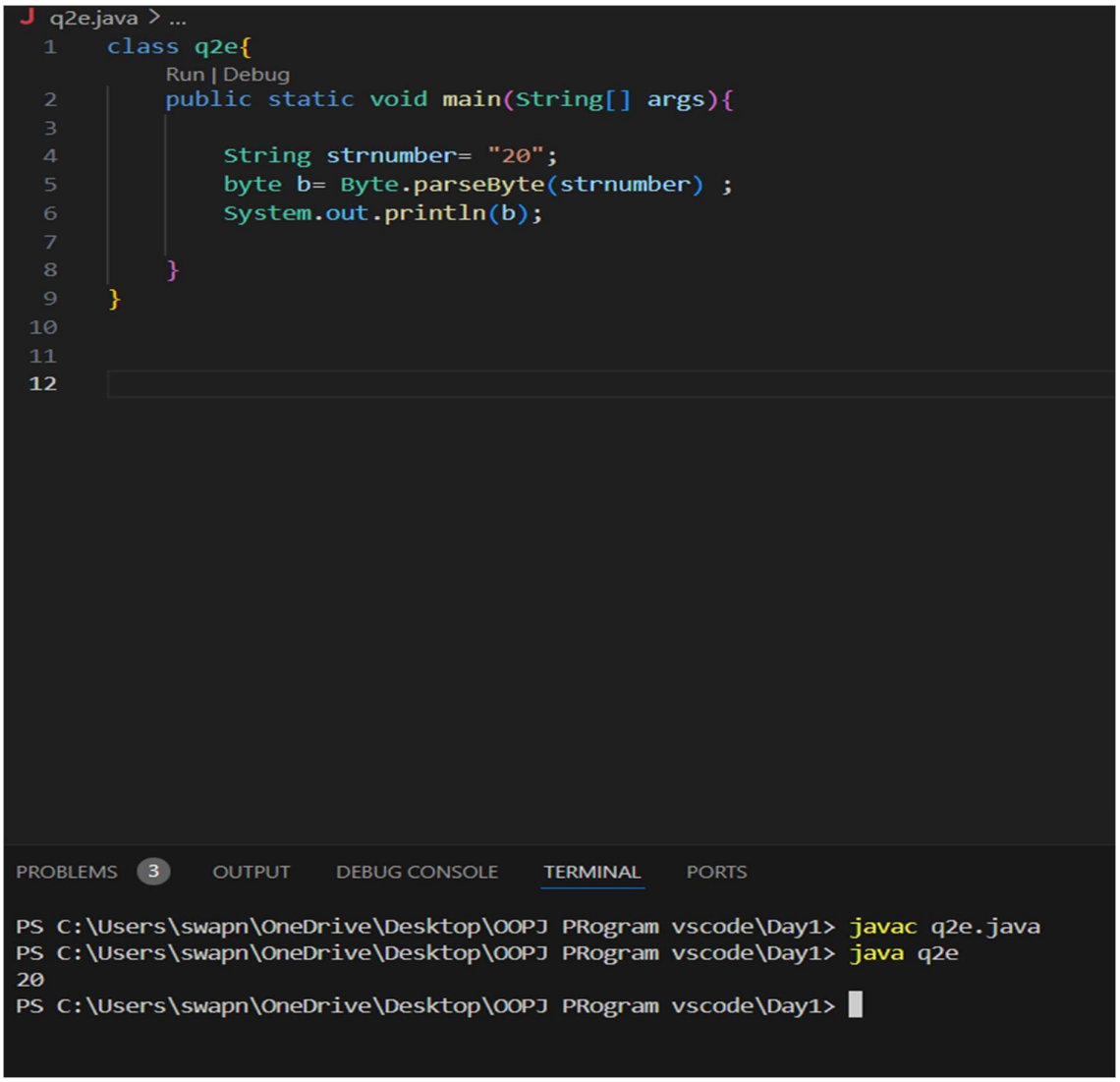
c. Write a program to find the minimum and maximum values of byte using the MIN_VALUE and MAX_VALUE fields.



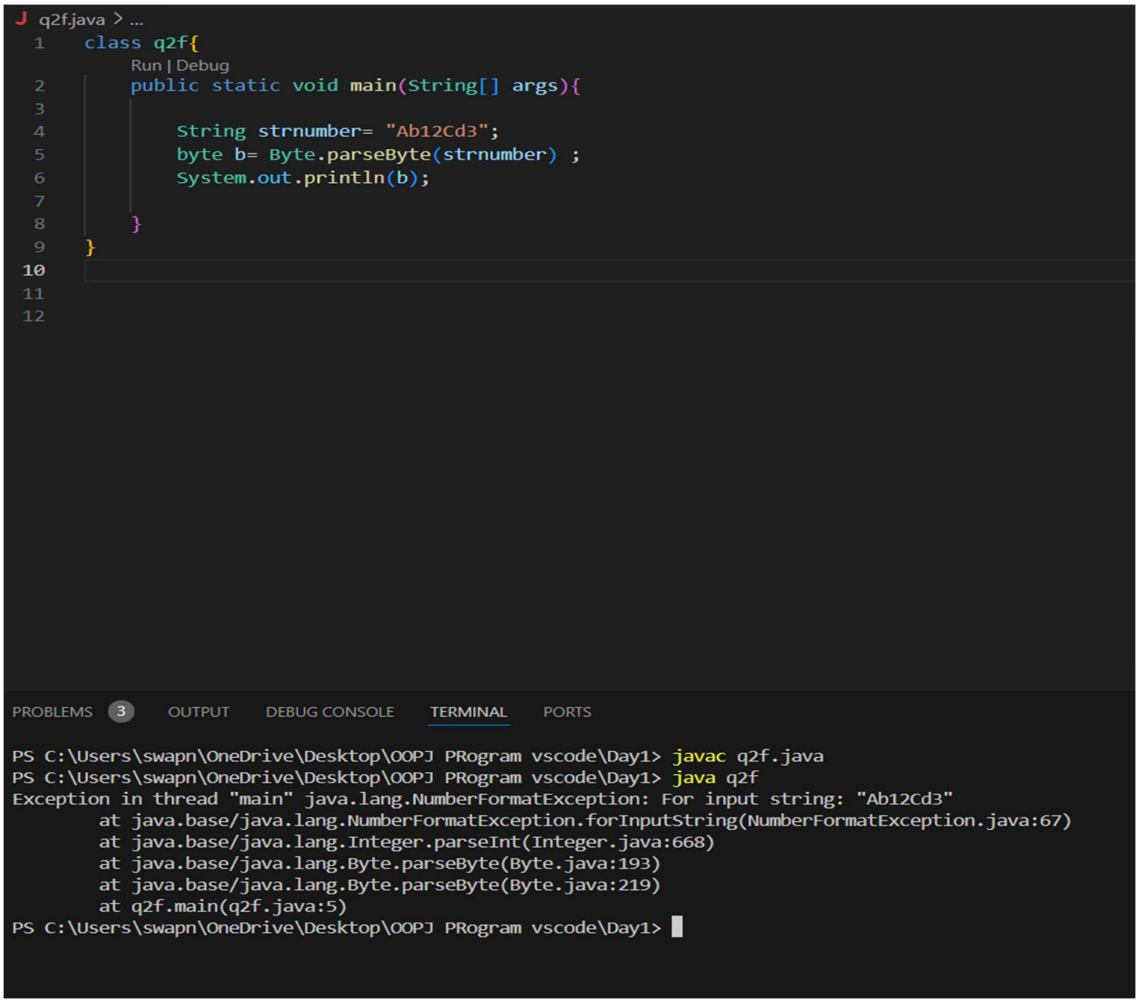
d. Declare a method-local variable number of type byte with some value and convert it to a String using the toString method.



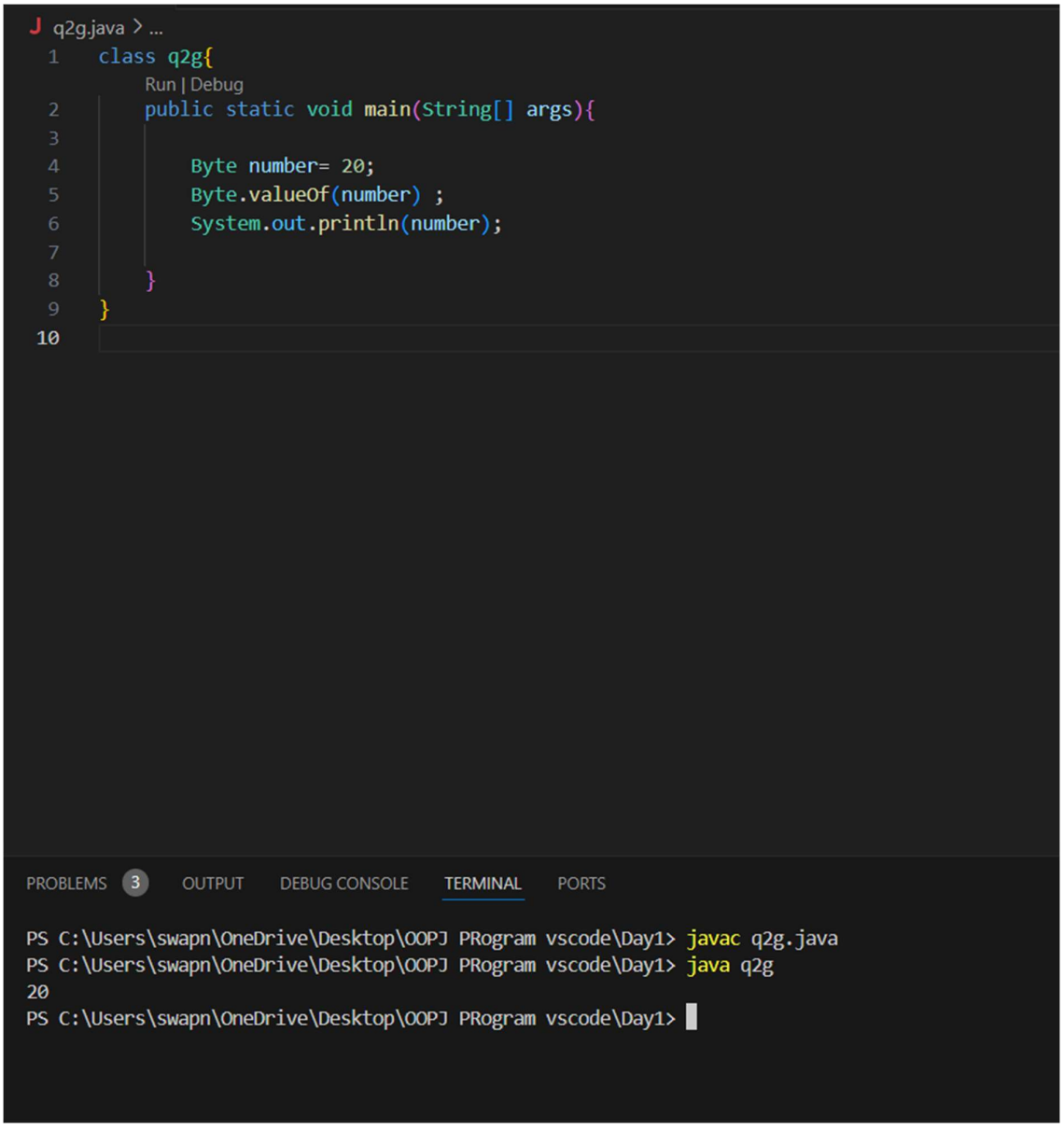
e. Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method.



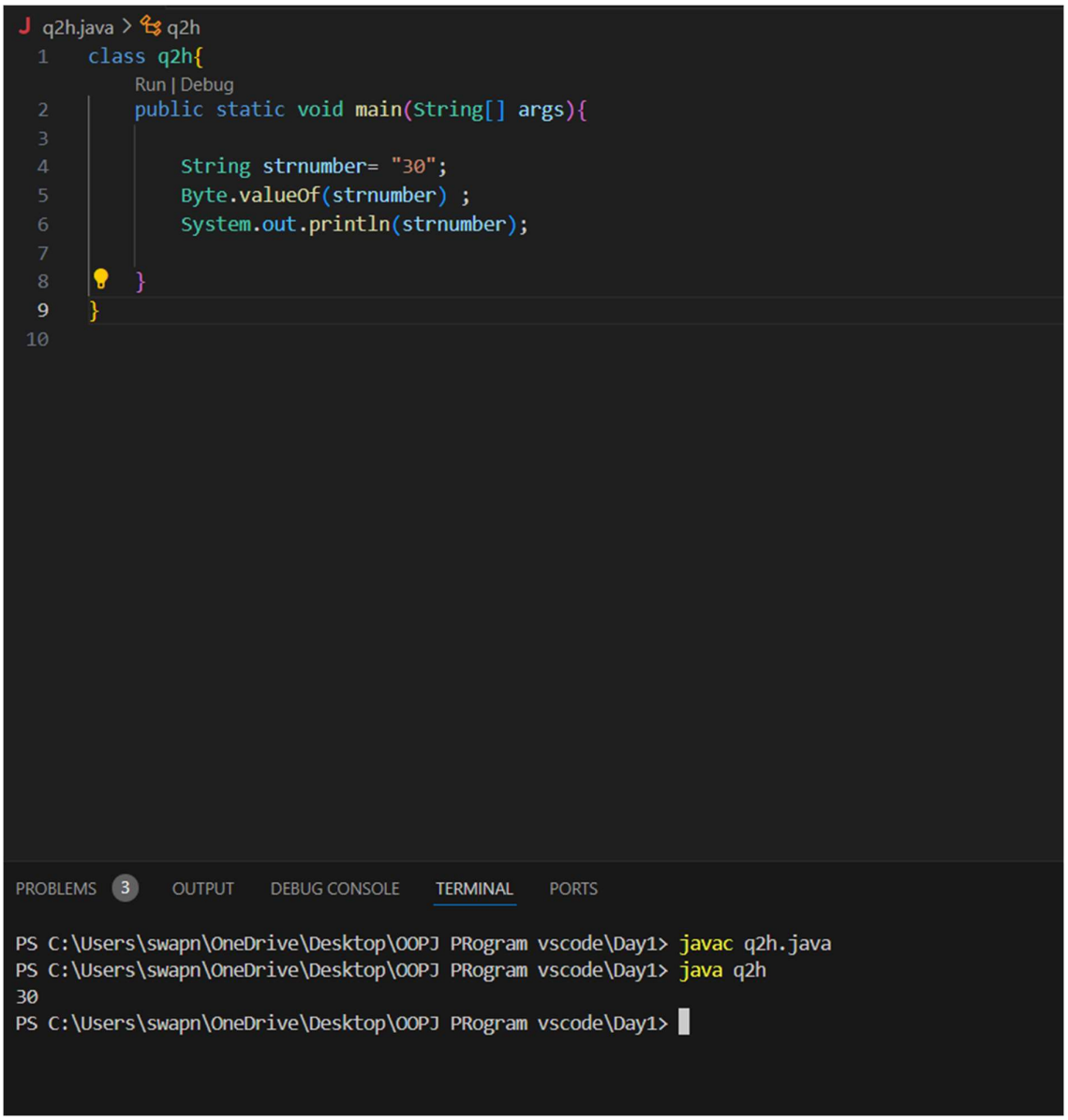
f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value.



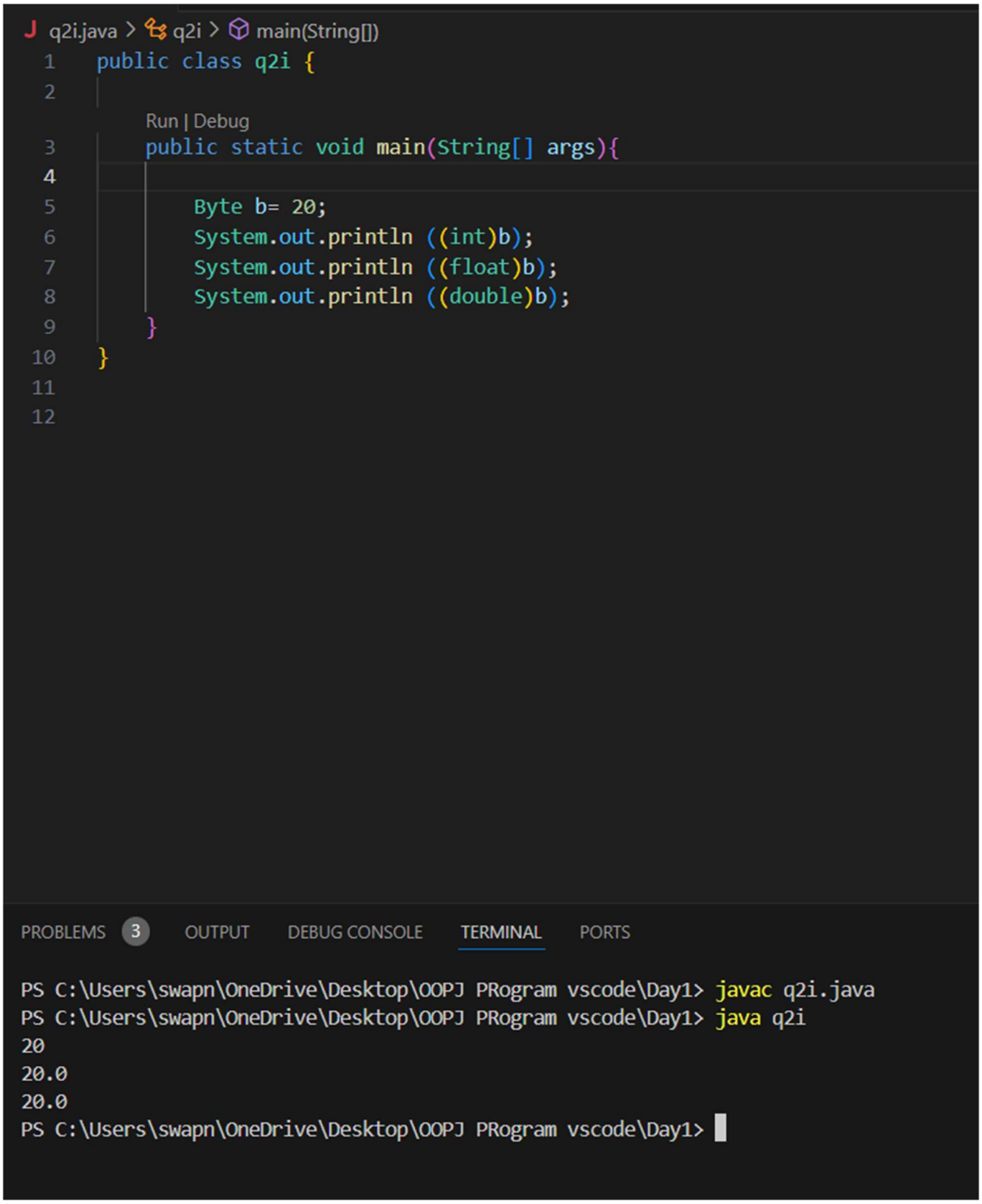
g. Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf().



h. Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf().



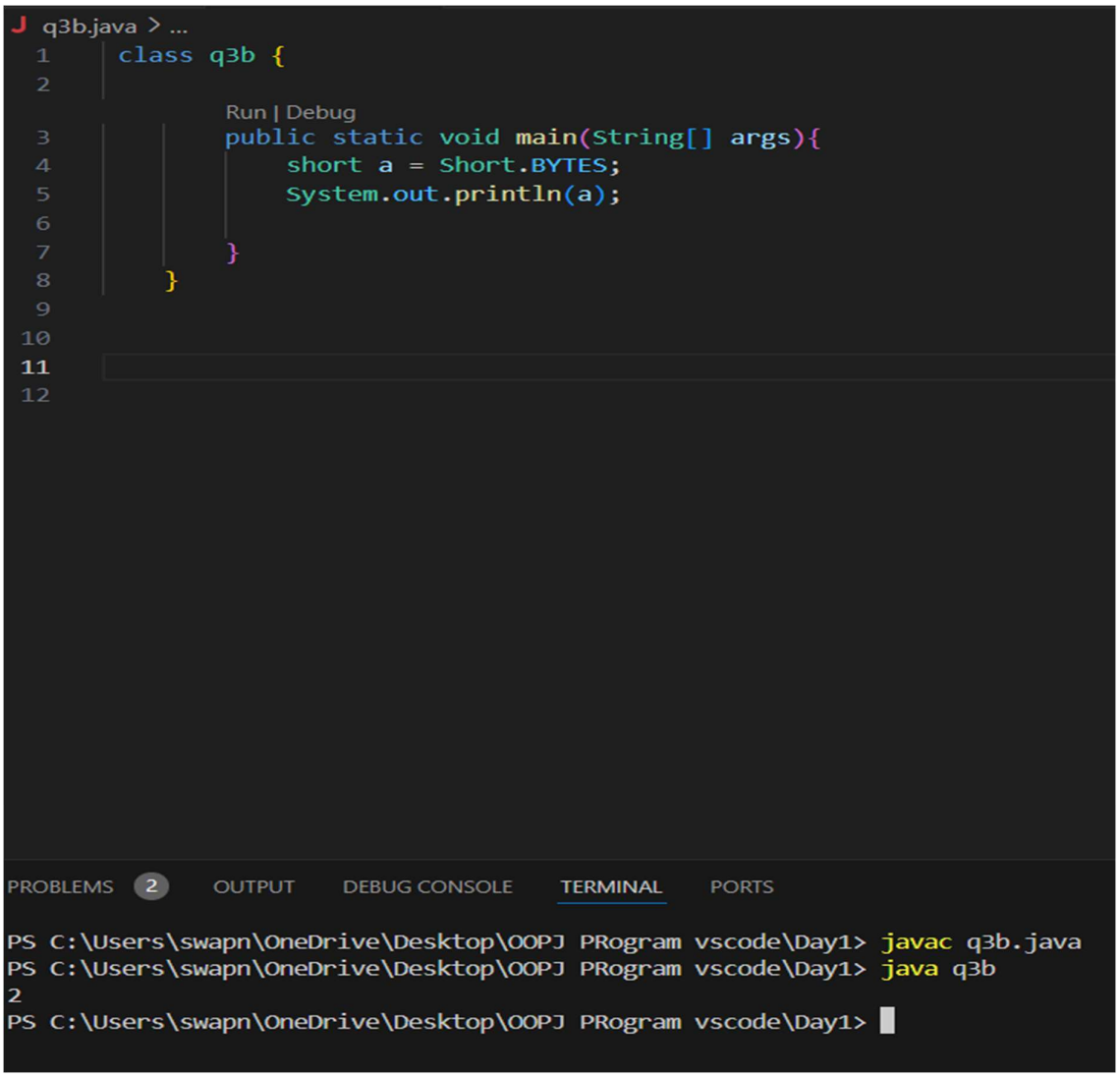
i. Experiment with converting a byte value into other primitive types or vice versa and observe the results.



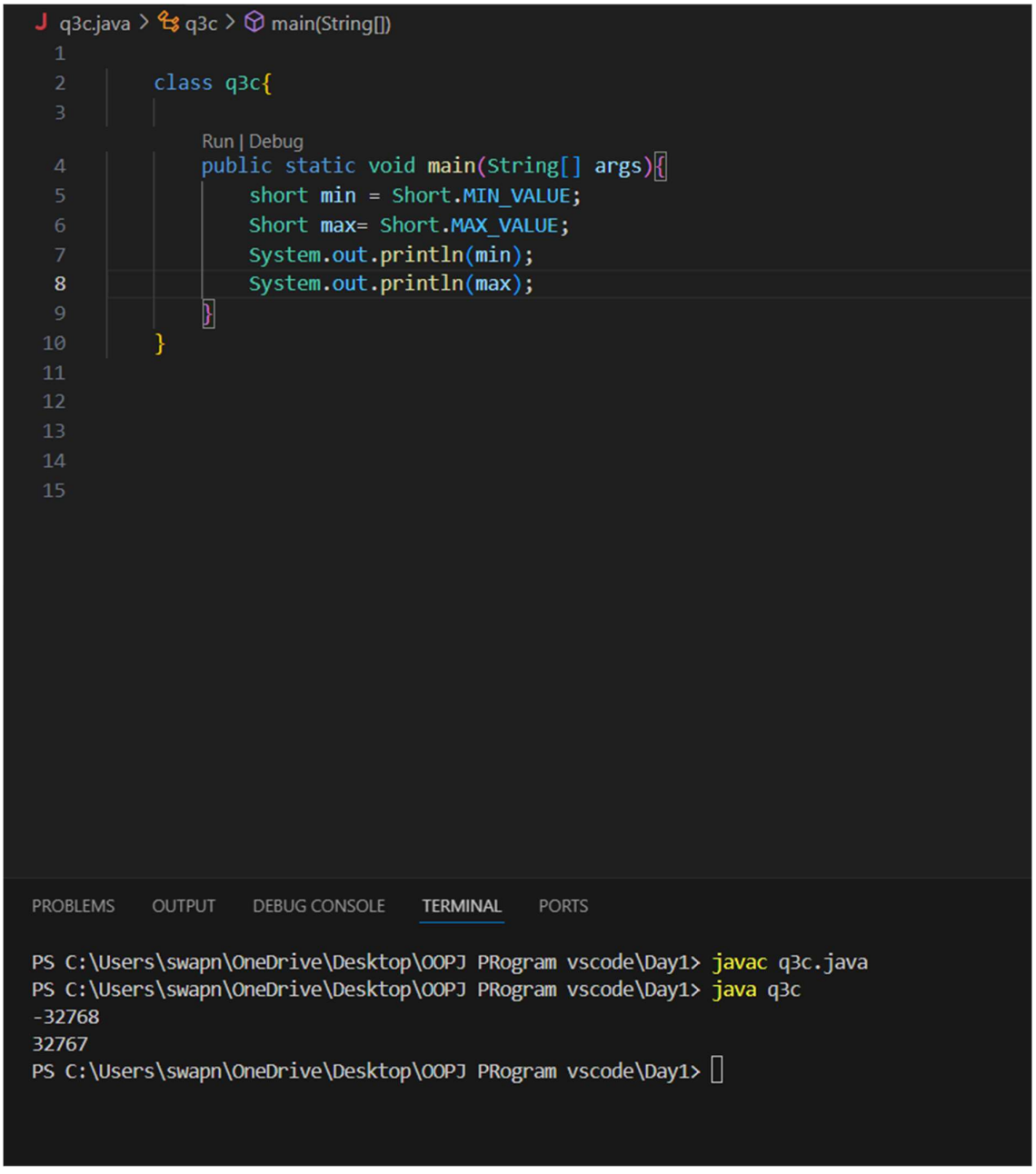
3. Working with java.lang.Short

a. Explore the Java API documentation for java.lang.Short and observe its modifiers and super types.

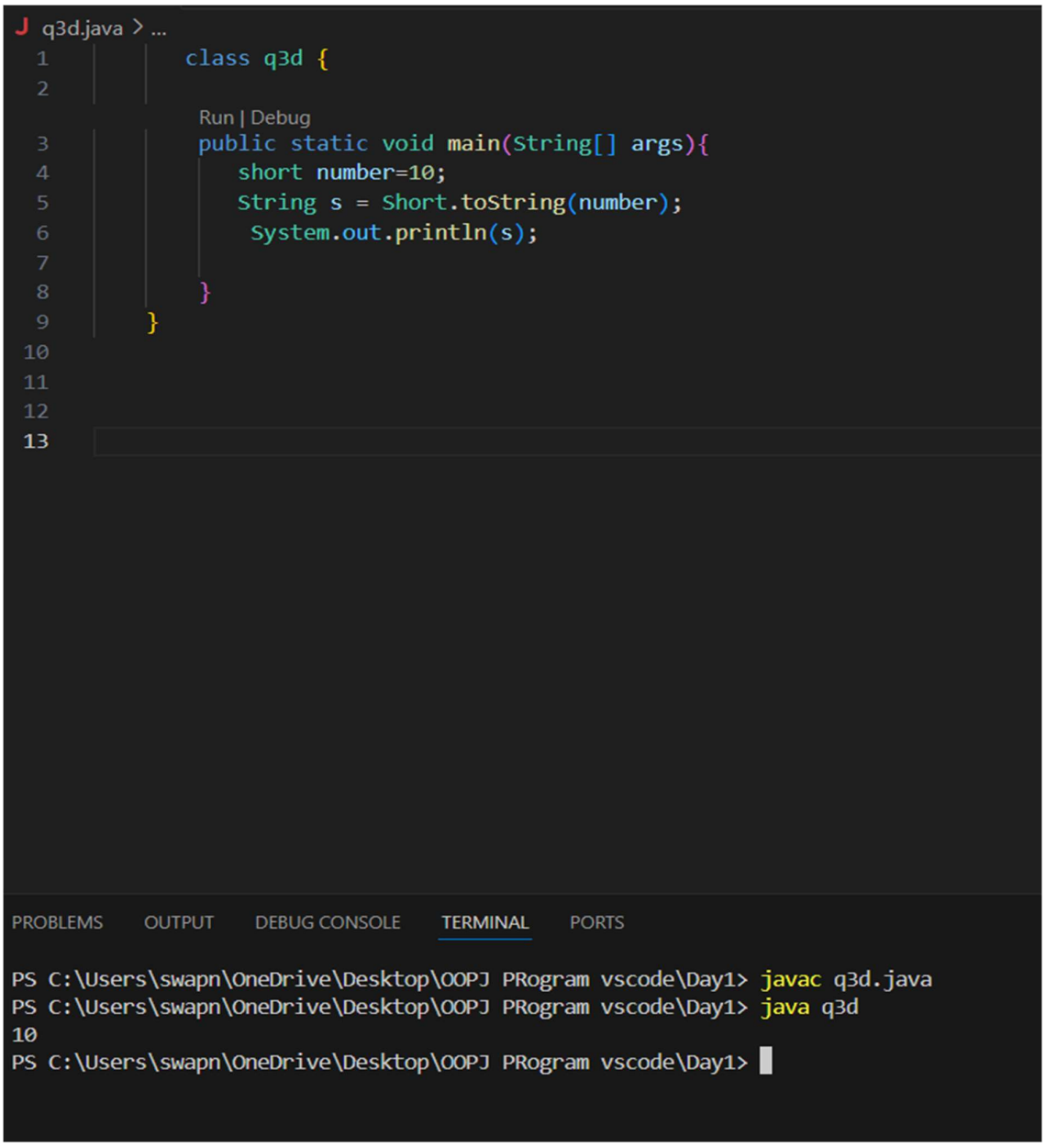
b. Write a program to test how many bytes are used to represent a short value using the BYTES field.



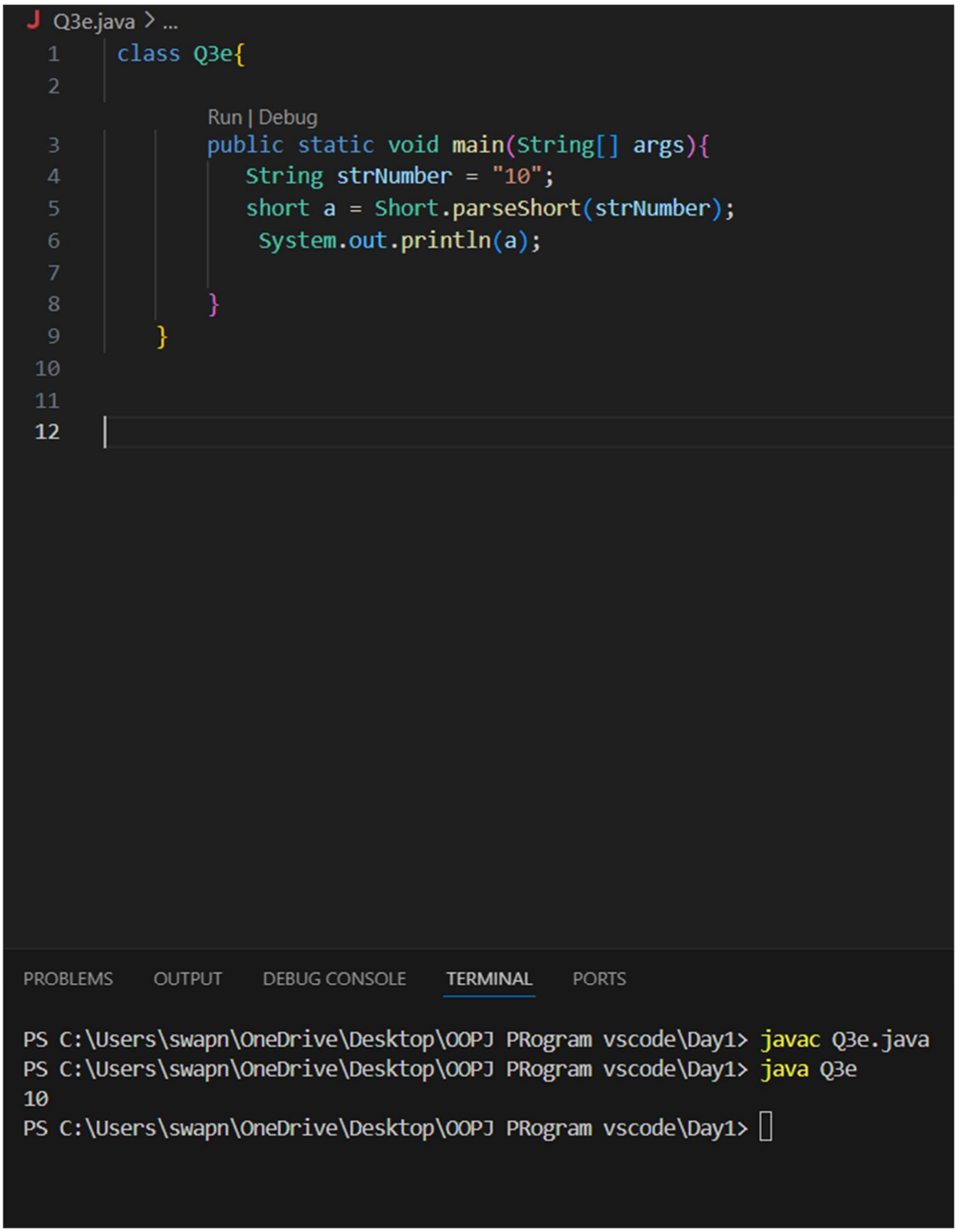
c. Write a program to find the minimum and maximum values of short using the MIN_VALUE and MAX_VALUE fields.



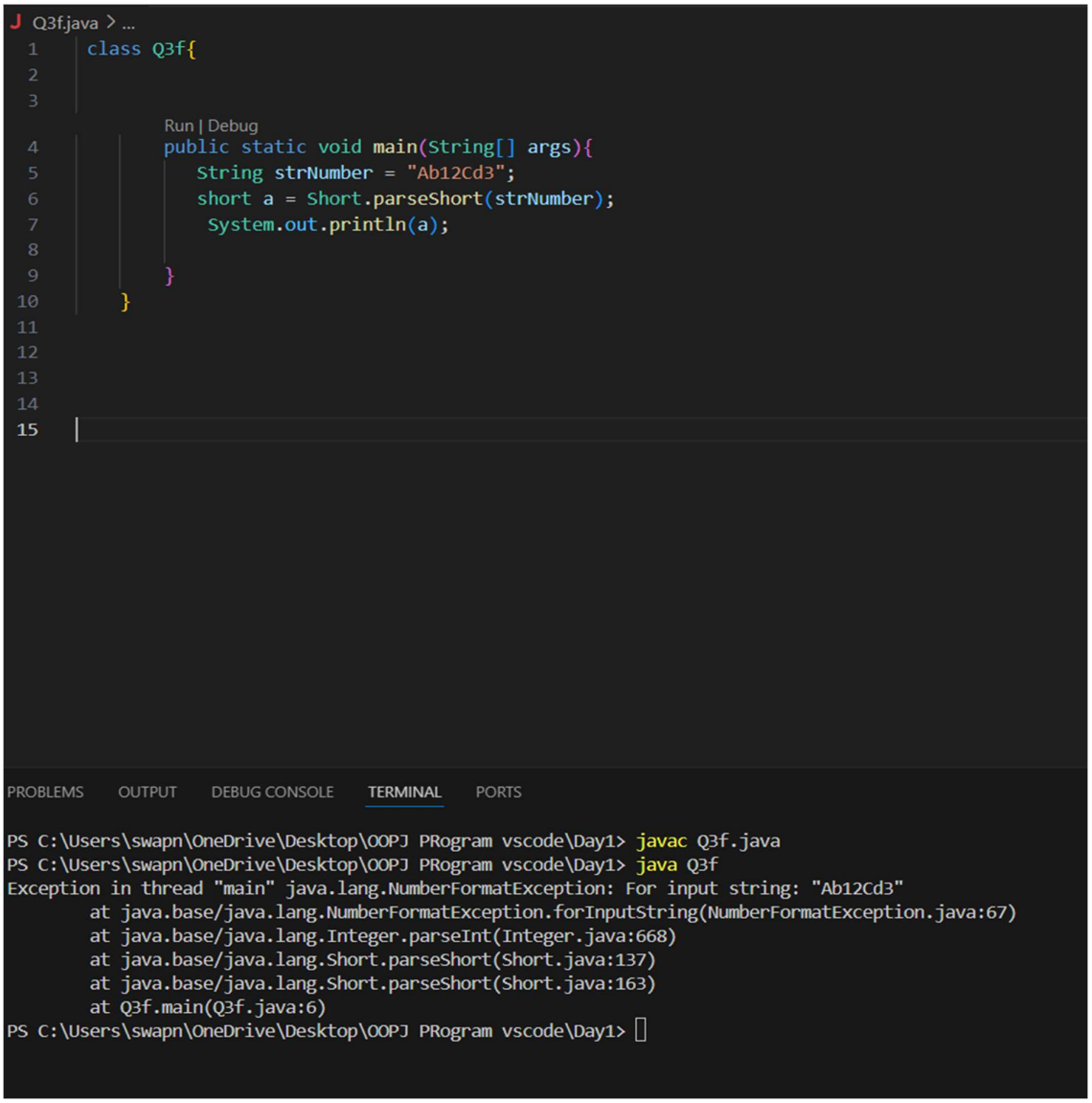
d. Declare a method-local variable number of type short with some value and convert it to a String using the toString method



e. Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method.

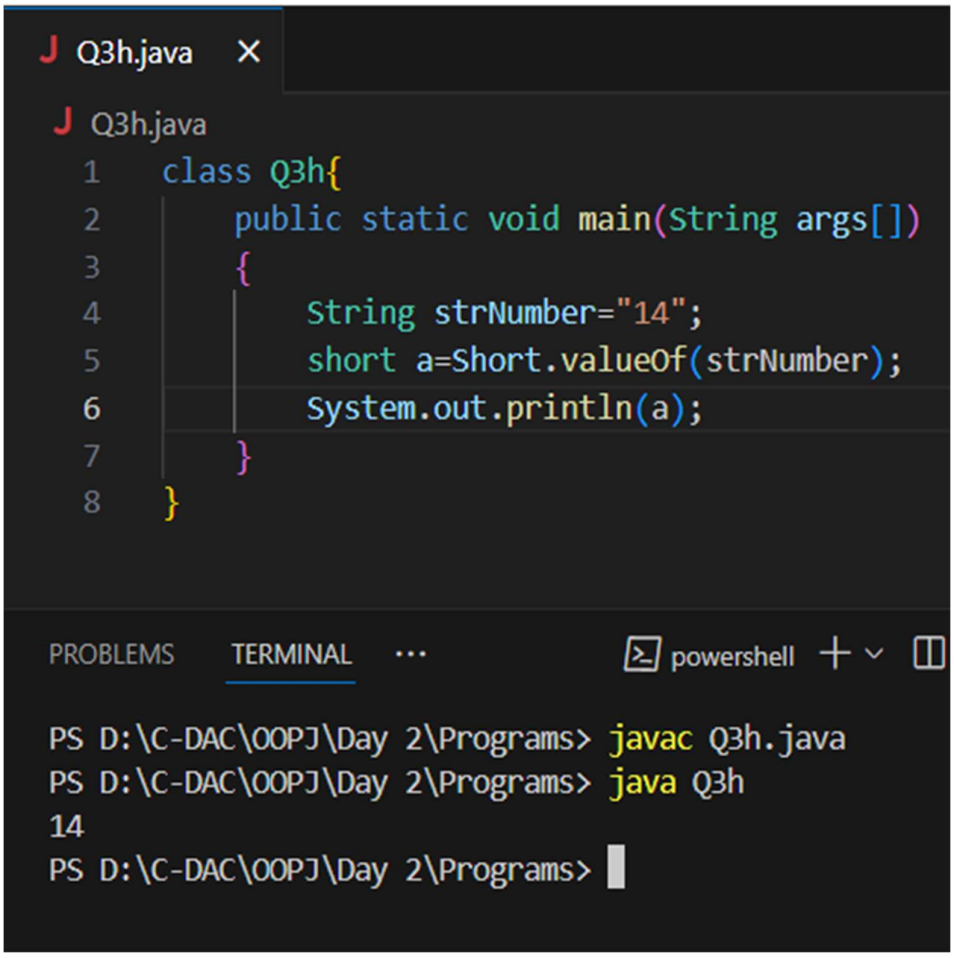


f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value.



g. Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf().

h. Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf().



i. Experiment with converting a short value into other primitive types or vice versa and observe the results.

Q3i.java > Q3i

```
1   class Q3i{
2   
        Run | Debug
3       public static void main(String[] args){
4           short a = 10;
5           System.out.println((int)a);
6           System.out.println((float)a);
7           System.out.println((long)a);
8           System.out.println((float)a);
9   
10          int b=20;
11          float c=25.05f;
12          double d = 10.00;
13          System.out.println((short)b);
14          System.out.println((short)c);
15      System.out.println((short)d);
16      }
17  
18  }
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q3i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q3i
10
10.0
10
10.0
20
25
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

4. Working with java.lang.Integer

a. Explore the Java API documentation for java.lang.Integer and observe its modifiers and super types.

b. Write a program to test how many bytes are used to represent an int value using the BYTES field.

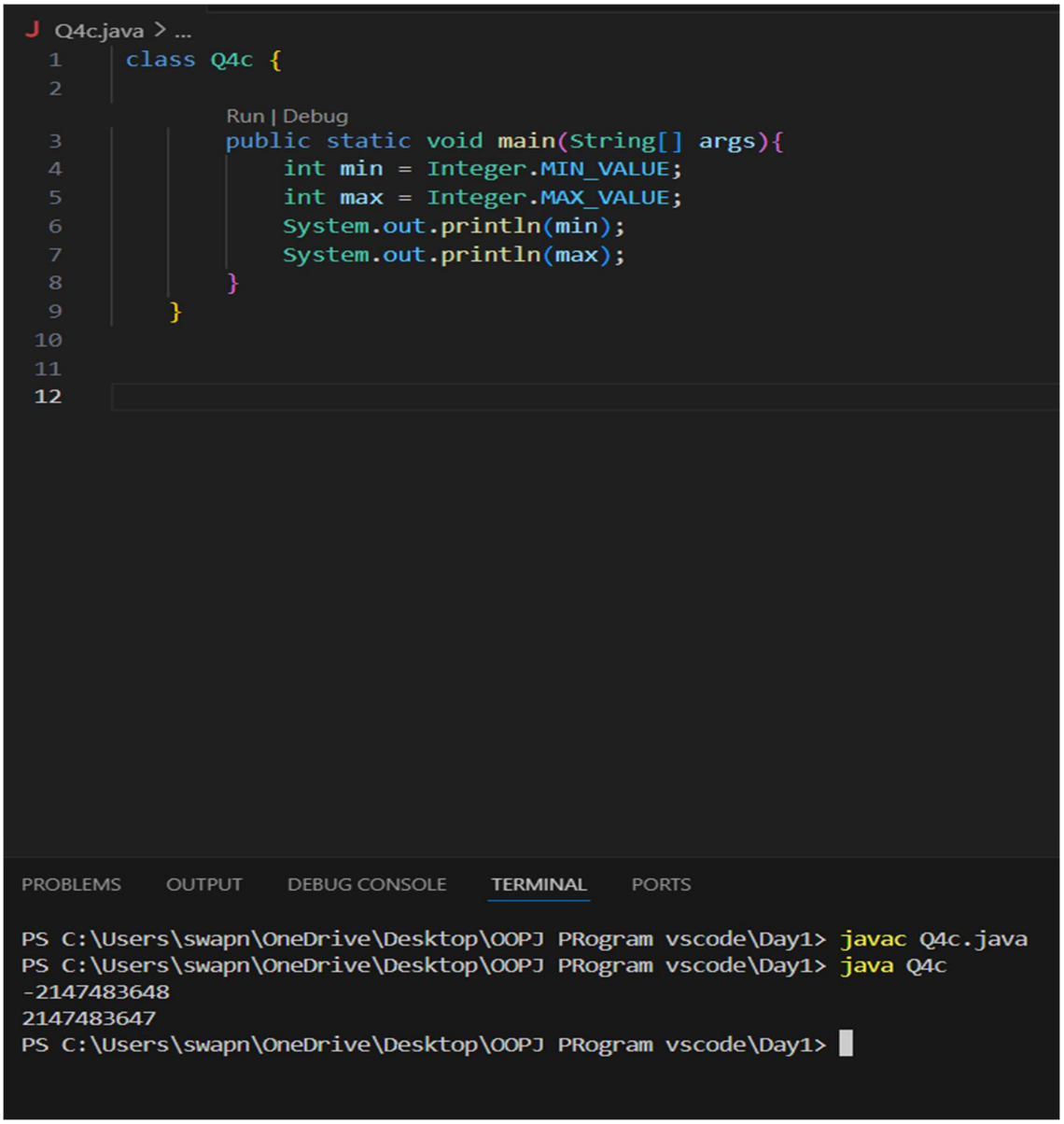
Q4b.java > ...

```java
public class Q4b {

    Run | Debug
    public static void main(String[] args){
        int a= Integer.BYTES;
        System.out.println(a);

    }
}

```

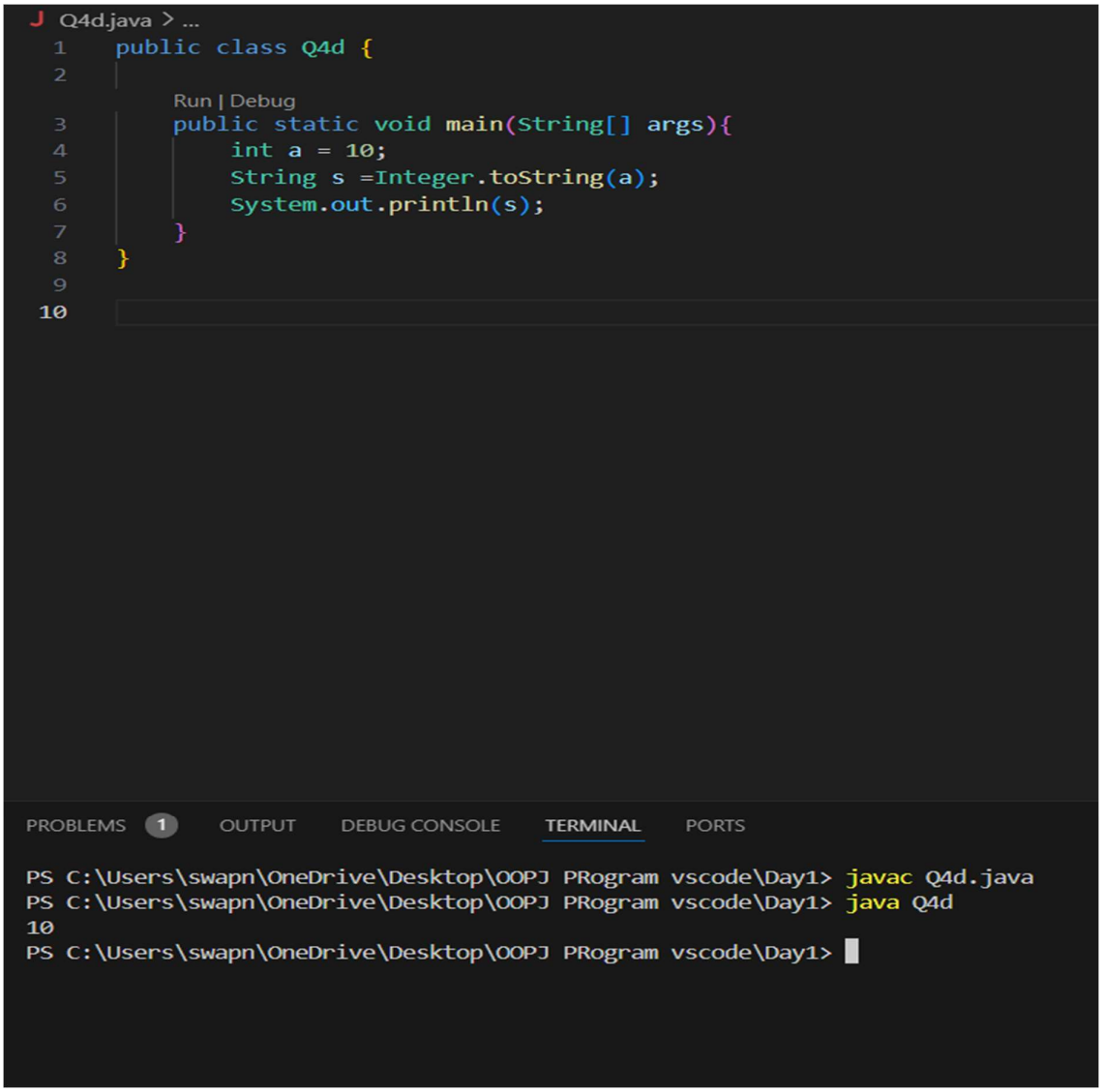
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4b
4
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

c. Write a program to find the minimum and maximum values of int using the MIN_VALUE and MAX_VALUE fields.

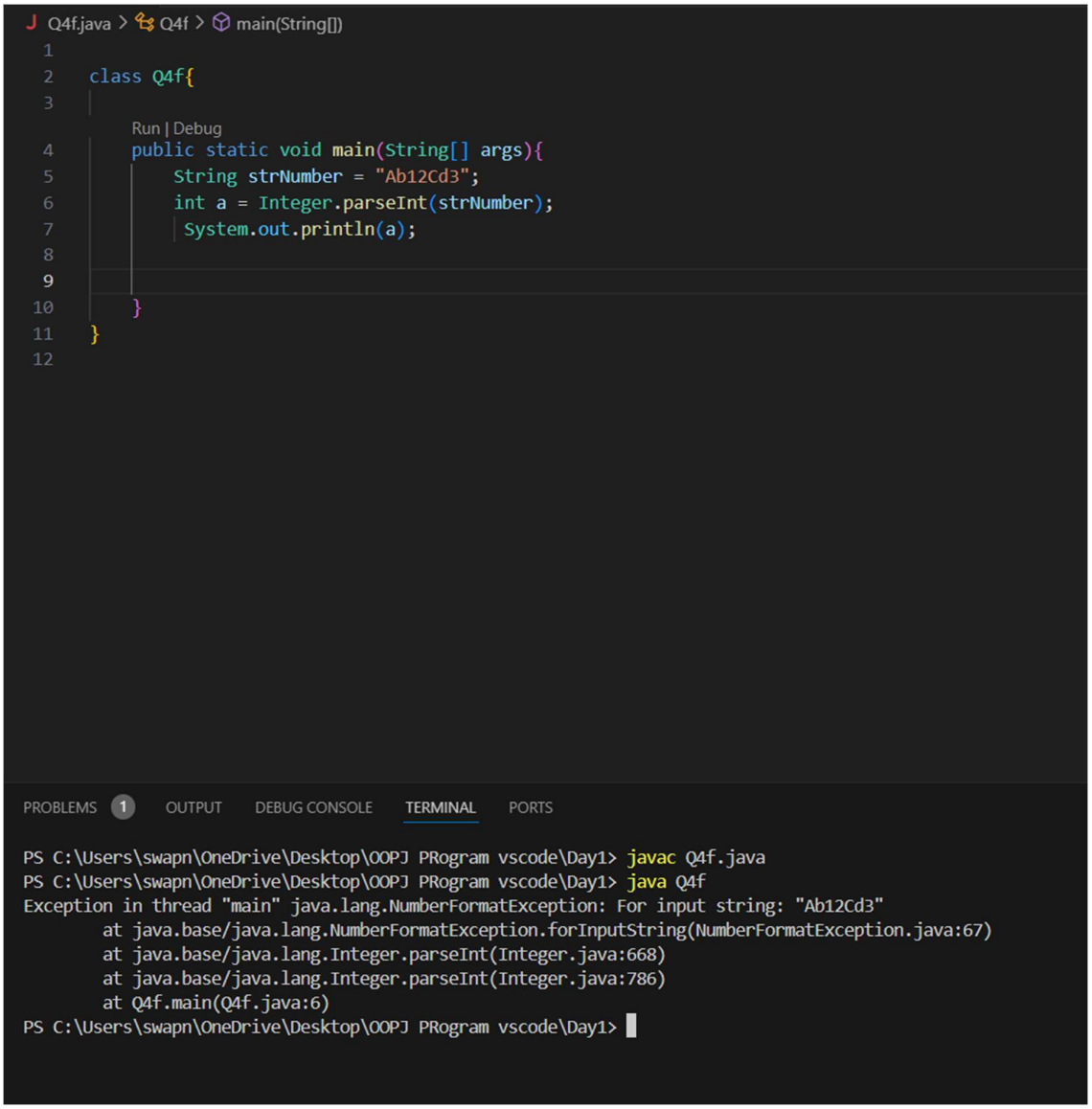


d. Declare a method-local variable number of type int with some value and convert it to a String using the toString method.



e. Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method.

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value.



g. Dclare a method-local variable number of type int with some value and convert it to the corresponding wrapper class using Integer.valueOf().

h. Declare a method-local variable strNumber of type String with some integer value and convert it to the corresponding wrapper class using Integer.valueOf().

i. Declare two integer variables with values 10 and 20, and add them using a method from the Integer class.

Q4i.java > Q4i > main(String[])

```java
public class Q4i {

    Run | Debug
    public static void main(String[] args){
        int a = 10;
        int b= 20;
       int res=Integer.sum(a,b);
    System.out.println(res);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4i
30
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

j. Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class.

Q4j.java > ...

```java
public class Q4j {
    Run | Debug
    public static void main(String[] args){
        int a = 10;
        int b= 20;
       int min=Integer.min(a,b);
       int max=Integer.max(a,b);
       System.out.println("Minimum:" +  min);
       System.out.println("Maximum:"  +  max);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4j
Minimum:10
Maximum:20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

k. Declare an integer variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Integer class.

Q4k.java > Q4k

```java
public class Q4k{

    Run | Debug
    public static void main(String[] args){
        int a = 7;

      System.out.println("Binary:"  +  Integer.toBinaryString (a));
      System.out.println("Octal:"   + Integer.toOctalString(a));
      System.out.println("Hexadecimal:"    +  Integer.toHexString(a));
      }
    }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4k.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4k
Binary:111
Octal:7
Hexadecimal:7
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

l. Experiment with converting an int value into other primitive types or vice versa and observe the results.

```java
public class Q4l{

    Run | Debug
    public static void main(String[] args){
        int a = 10;
        int b= 40;
        float i =20.0f;
      System.out.println((float)a);
      System.out.println((double)a);
      System.out.println((long)a);
      System.out.println((float)b);
      System.out.println((double)b);
      System.out.println((long)b);
      System.out.println((long)i);
      System.out.println((double)i);
      System.out.println((short)i);
      }
    }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4l.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4l
10.0
10.0
10
40.0
40.0
40
20
20.0
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

5. Working with java.lang.Long

a. Explore the Java API documentation for java.lang.Long and observe its modifiers and super types.

b. Write a program to test how many bytes are used to represent a long value using the BYTES field.

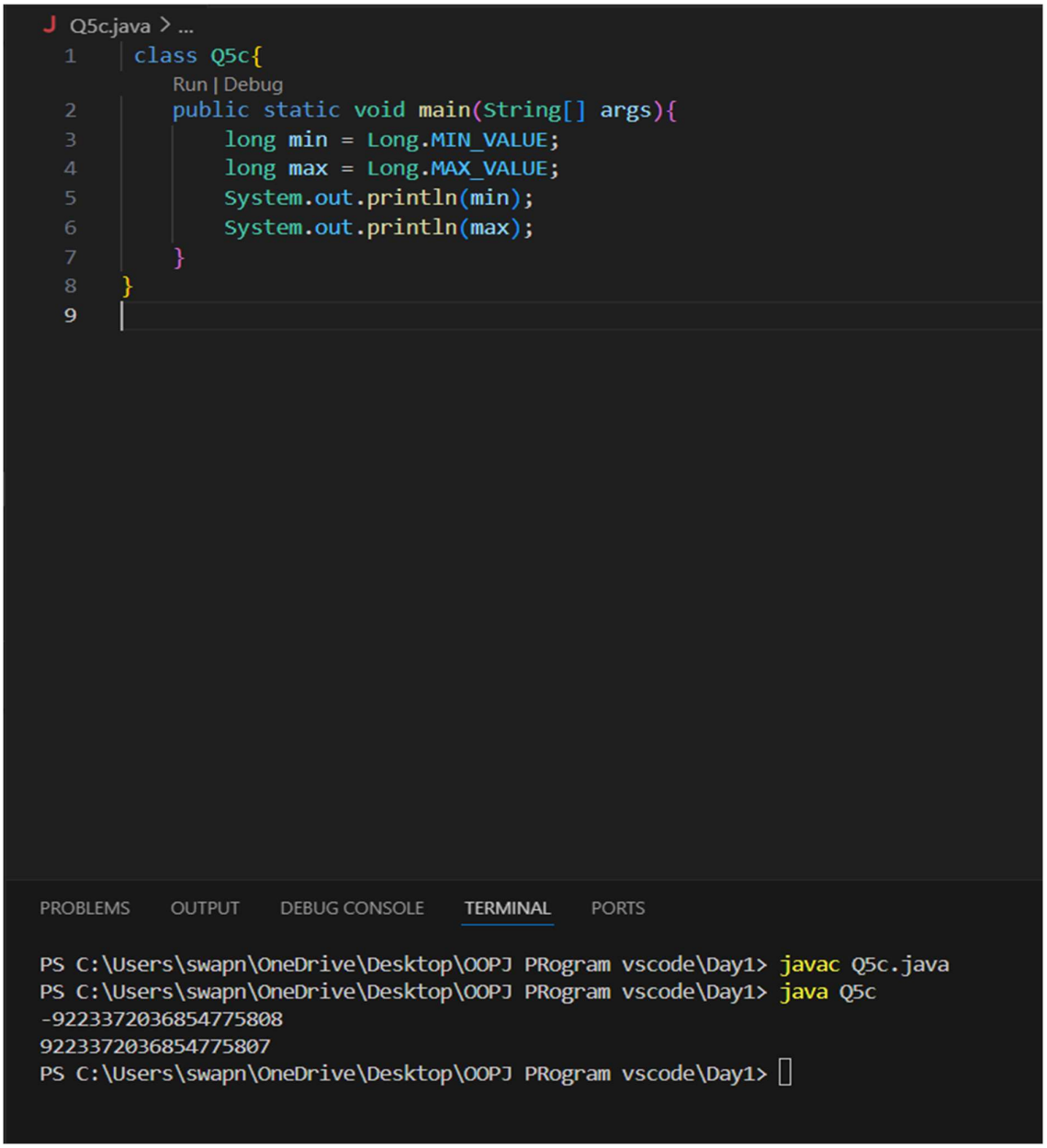
Q5b.java > ...

```java
public class Q5b {

    Run | Debug
    public static void main(String[] args){
     long a = Long.BYTES;
     System.out.println(a);

    }
}

```

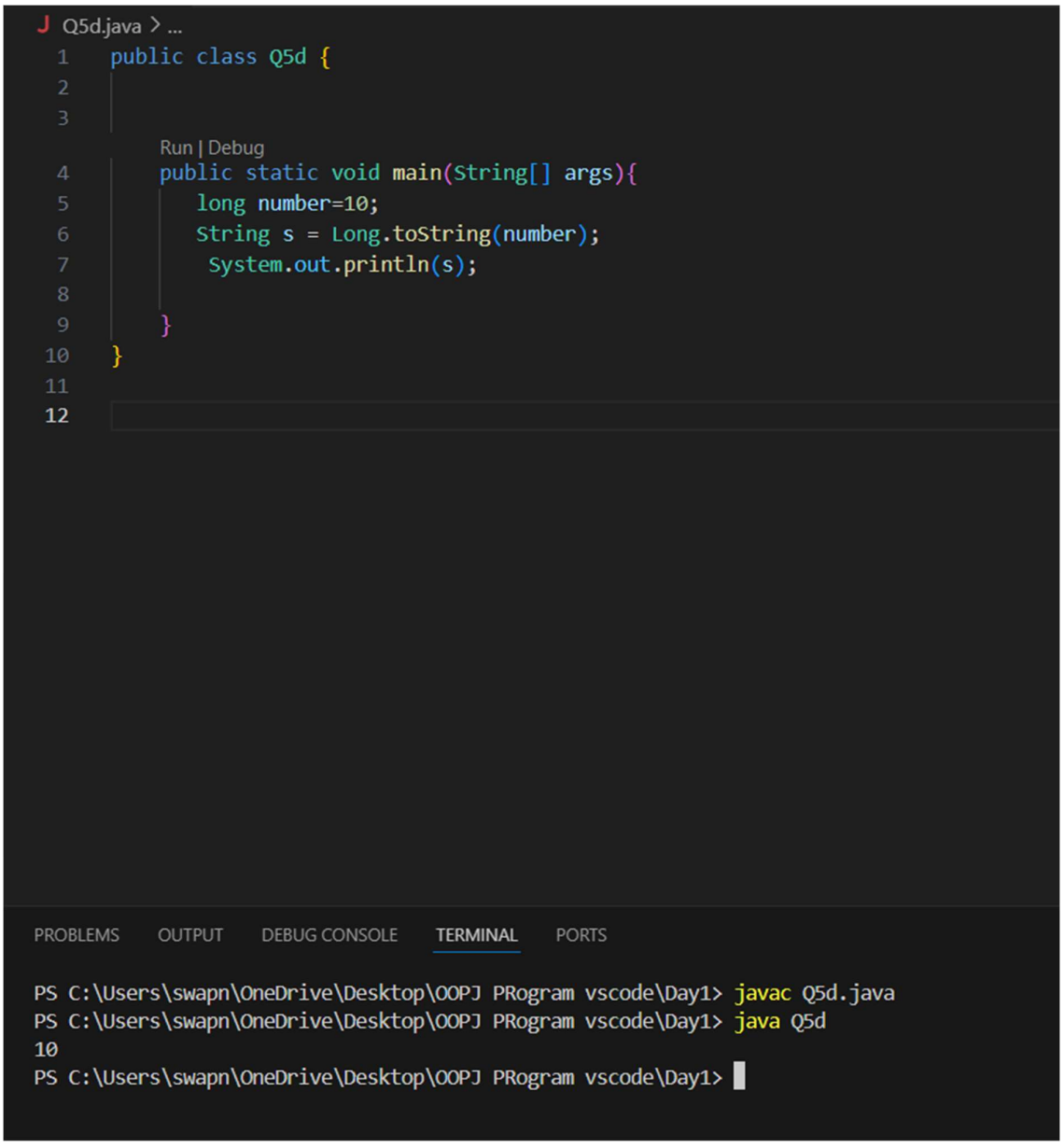
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5b
8
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

c.Write a program to find the minimum and maximum values of long using the MIN_VALUE and MAX_VALUE fields.

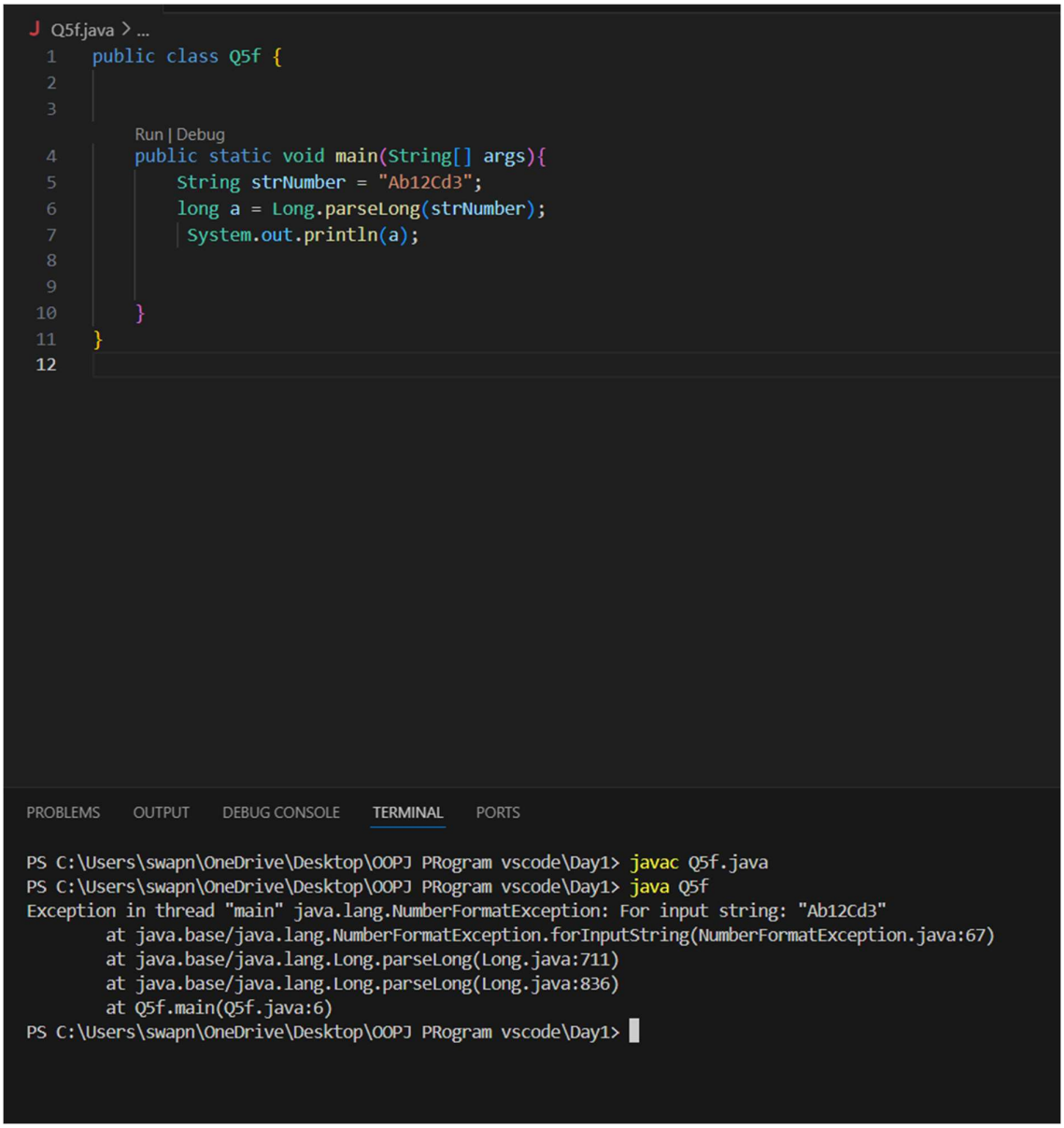


d. Declare a method-local variable number of type long with some value and convert it to a String using the toString method.

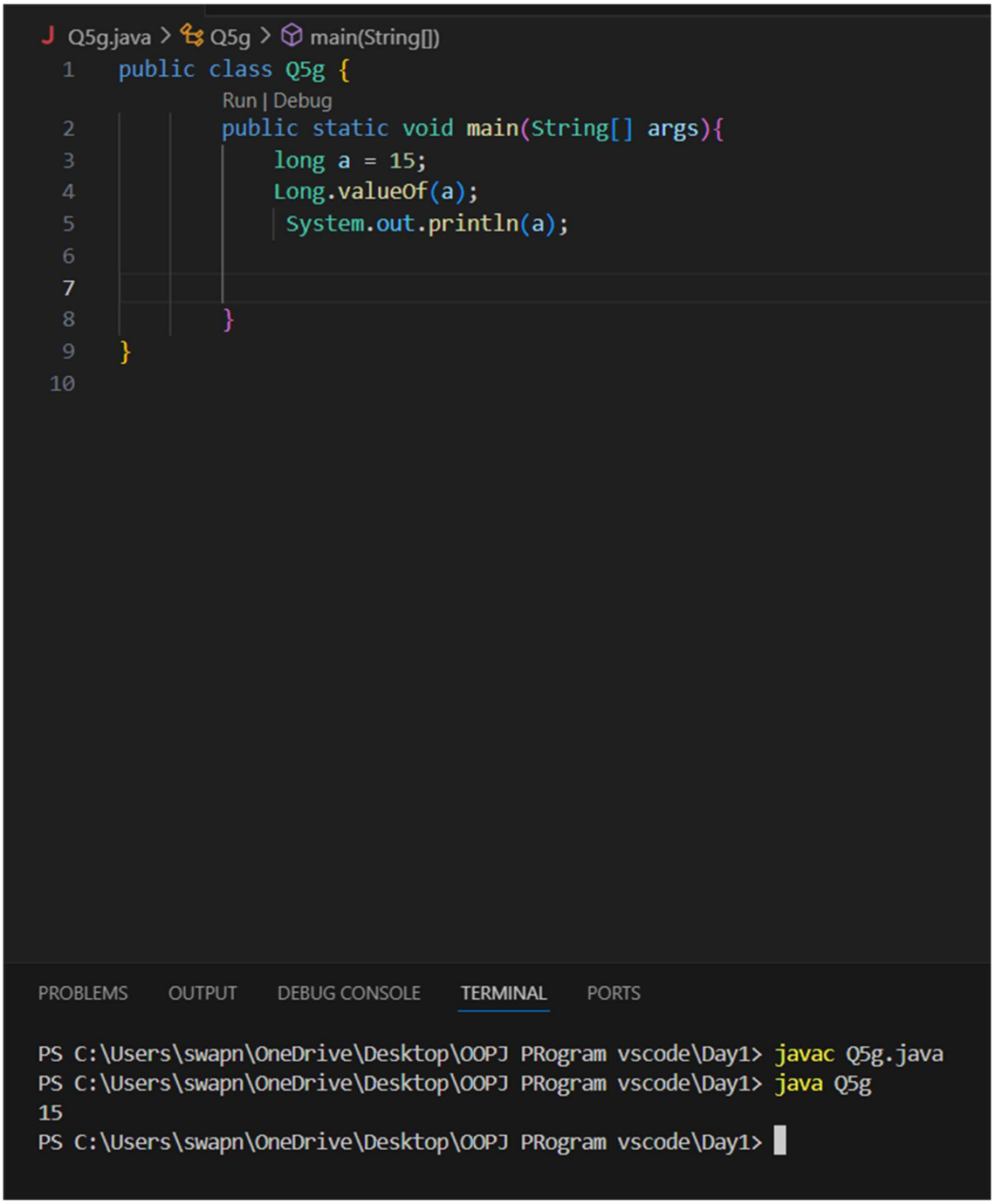


e. Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method.

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value.

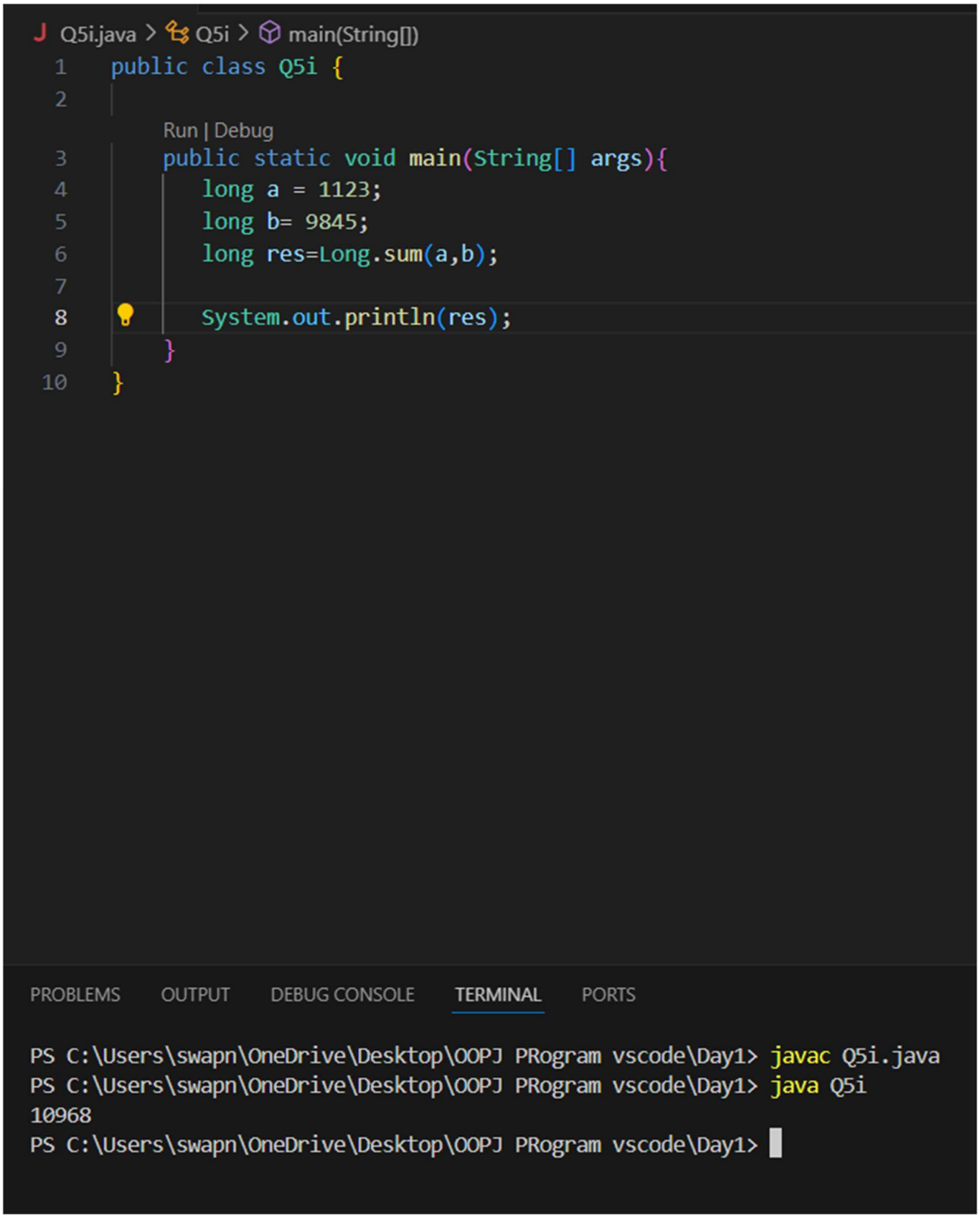


g. Declare a method-local variable number of type long with some value and convert it to the corresponding wrapper class using Long.valueOf().



h. Declare a method-local variable strNumber of type String with some long value and convert it to the corresponding wrapper class using Long.valueOf().

i.Declare two long variables with values 1123 and 9845, and add them using a method from the Long class.



j. Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class.

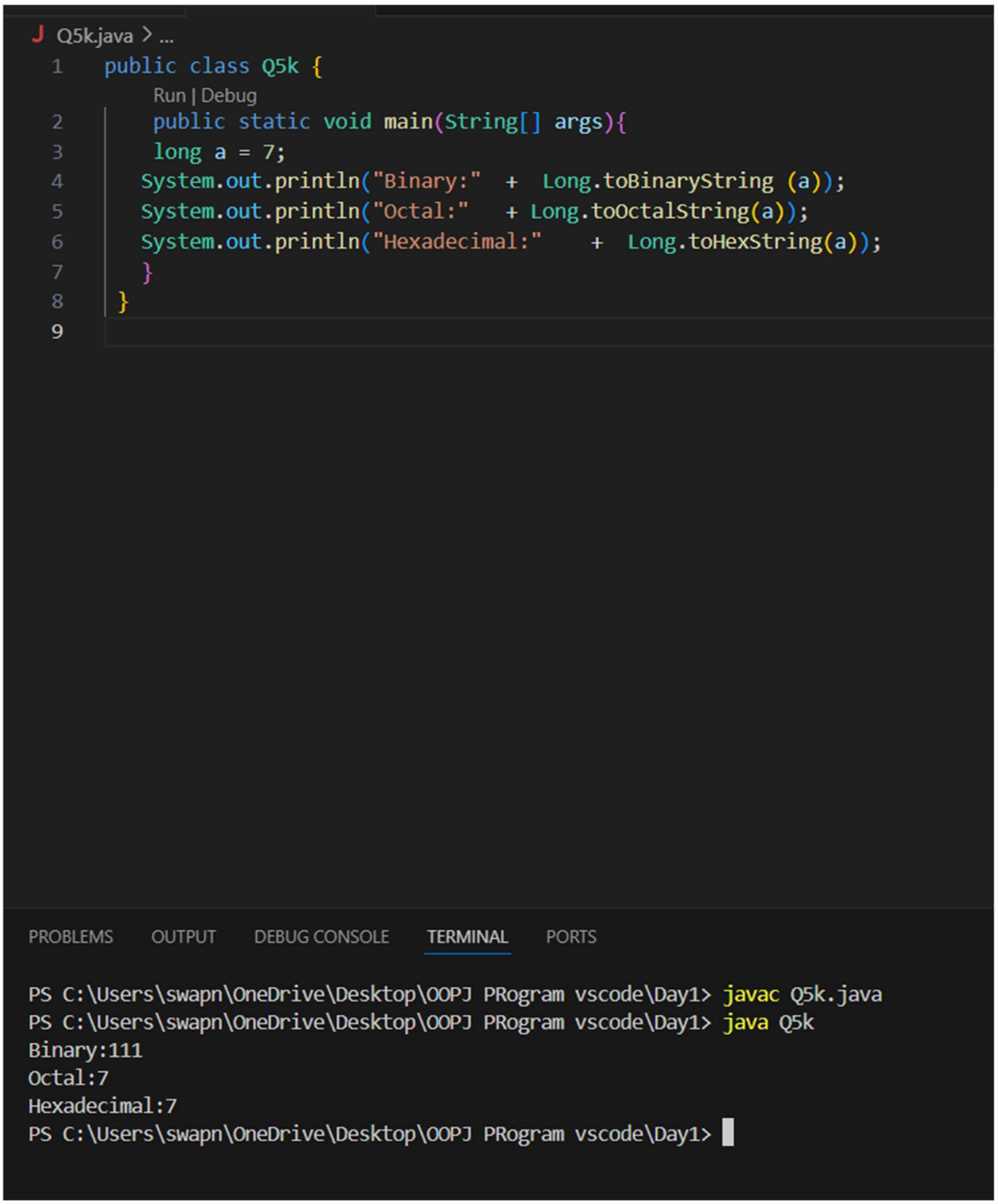
Q5j.java > ...

```java
public class Q5j {

    Run | Debug
    public static void main(String[] args){
        long a = 1122;
        long b= 5566;
        long min=Long.min(a,b);
        long max=Long.max(a,b);
       System.out.println("Minimum:" +  min);
       System.out.println("Maximum:"  +  max);
    }
}

```

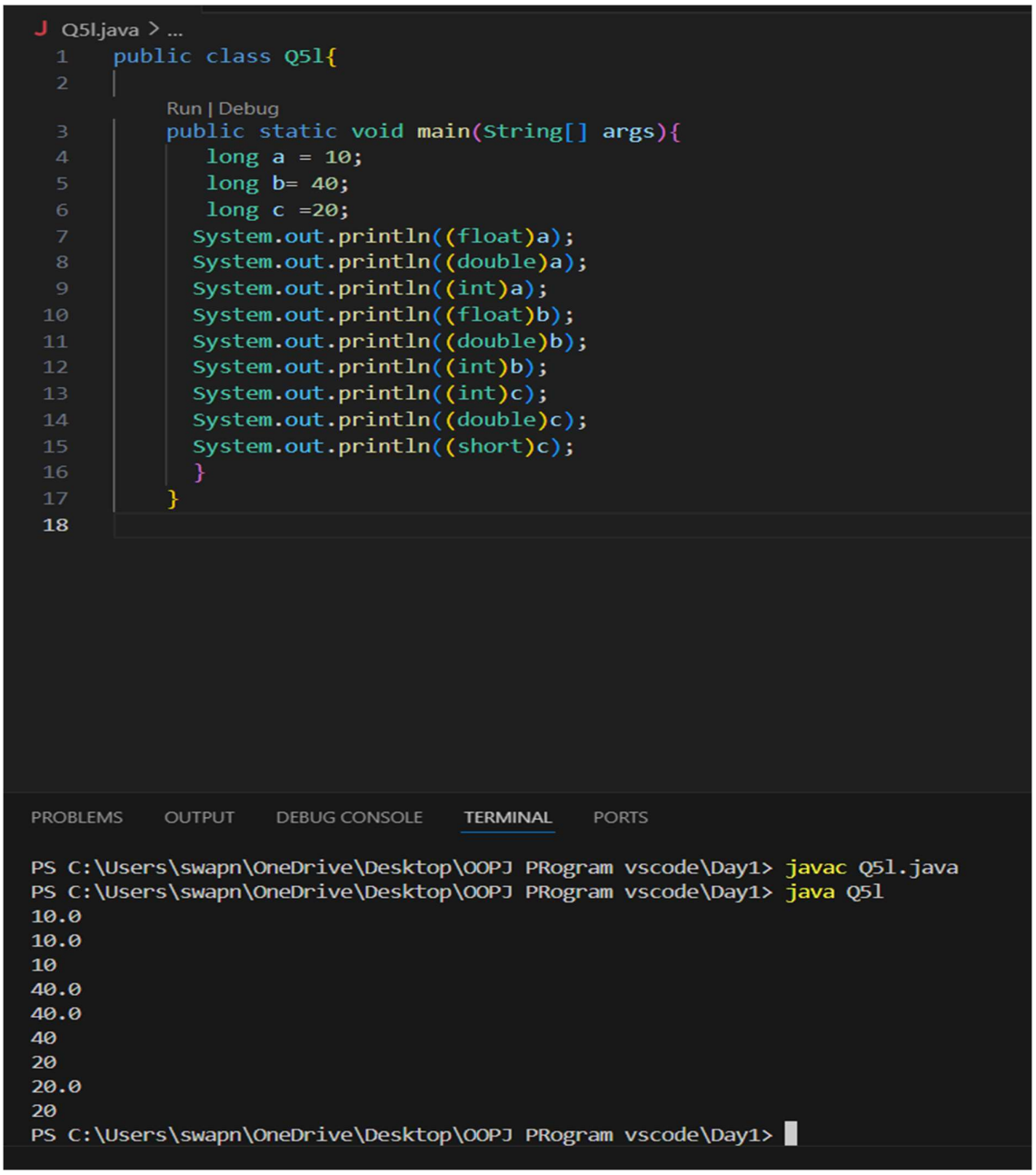
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5j
Minimum:1122
Maximum:5566
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

k. Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class.



l. Experiment with converting a long value into other primitive types or vice versa and observe the results.



```
Q5l.java > ...
1   public class Q5l{
2   
        Run | Debug
3       public static void main(String[] args){
4          long a = 10;
5          long b= 40;
6          long c =20;
7         System.out.println((float)a);
8         System.out.println((double)a);
9         System.out.println((int)a);
10        System.out.println((float)b);
11        System.out.println((double)b);
12        System.out.println((int)b);
13        System.out.println((int)c);
14        System.out.println((double)c);
15        System.out.println((short)c);
16        }
17      }
18
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5l.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5l
10.0
10.0
10
40.0
40.0
40
20
20.0
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

6. Working with java.lang.Float

a. Explore the Java API documentation for java.lang.Float and observe its modifiers and super types.

b. Write a program to test how many bytes are used to represent a float value using the BYTES field.

q6b.java

```java
class q6b{
Run | Debug
public static void main(String[] args){
float a =Float.BYTES;
System.out.println(a);
}
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6b
4.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

c. Write a program to find the minimum and maximum values of float using the MIN_VALUE and MAX_VALUE fields. ss

d. Declare a method-local variable number of type float with some value and convert it to a String using the toString method

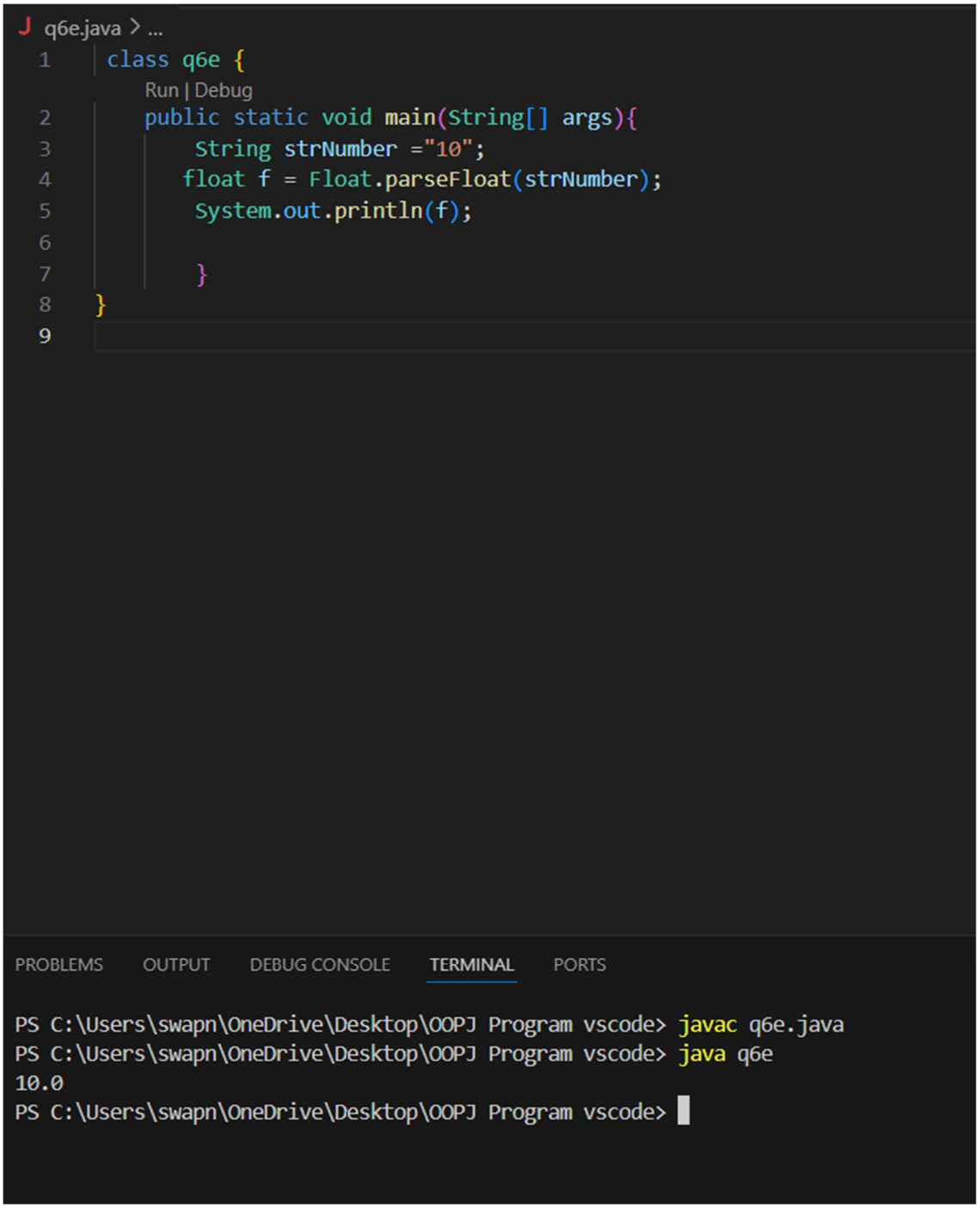
q6d.java > q6d

```java
class q6d {

    Run | Debug
    public static void main(String[] args){
    float number = 10f;
   String s = Float.toString(number);
    System.out.println(s);
    }
    }
```

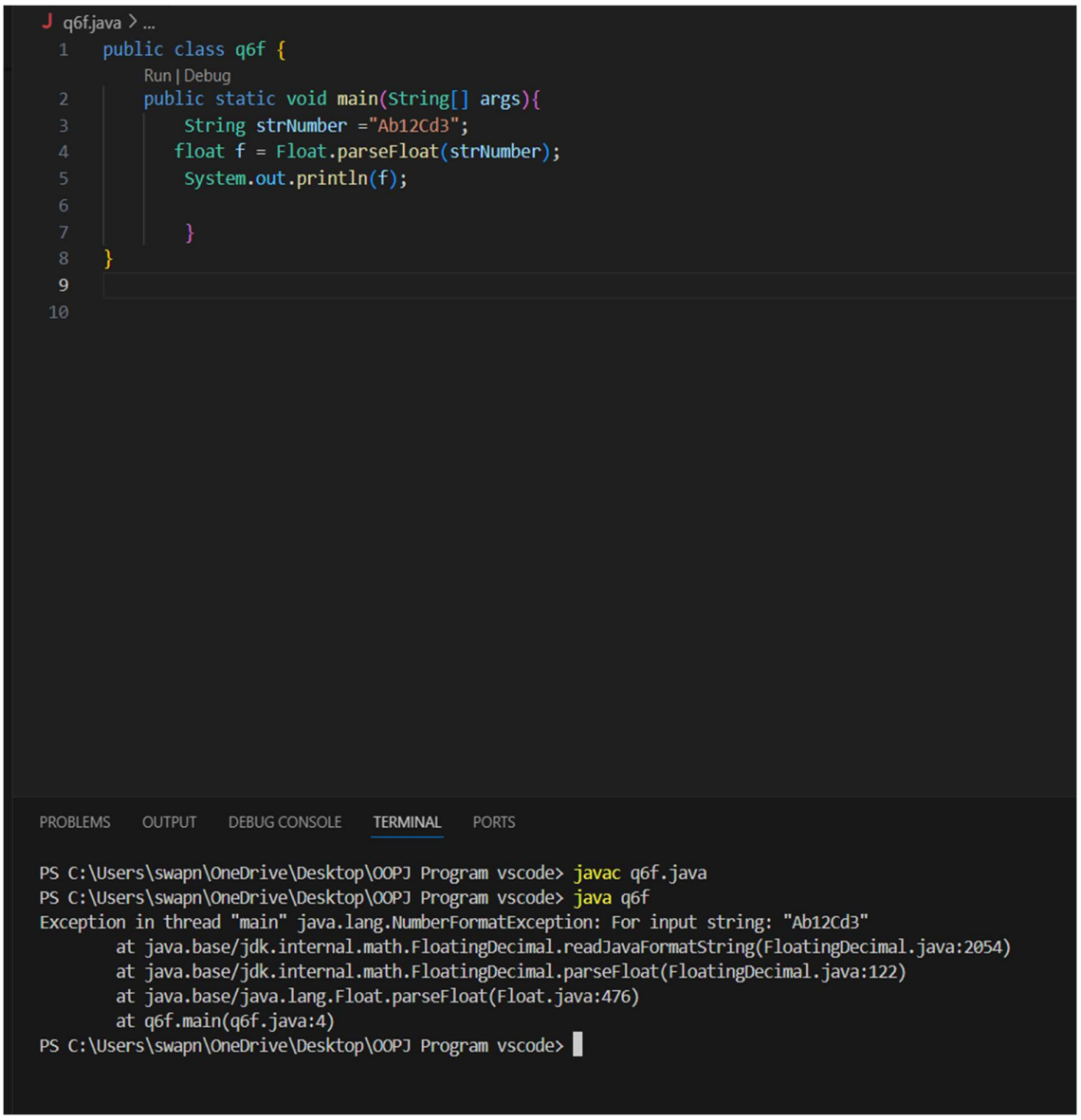
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6d
10.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a float value using the parseFloat method.



f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a float value.



g. Declare a method-local variable number of type float with some value and convert it to the corresponding wrapper class using Float.valueOf().

h. Declare a method-local variable strNumber of type String with some float value and convert it to the corresponding wrapper class using Float.valueOf().

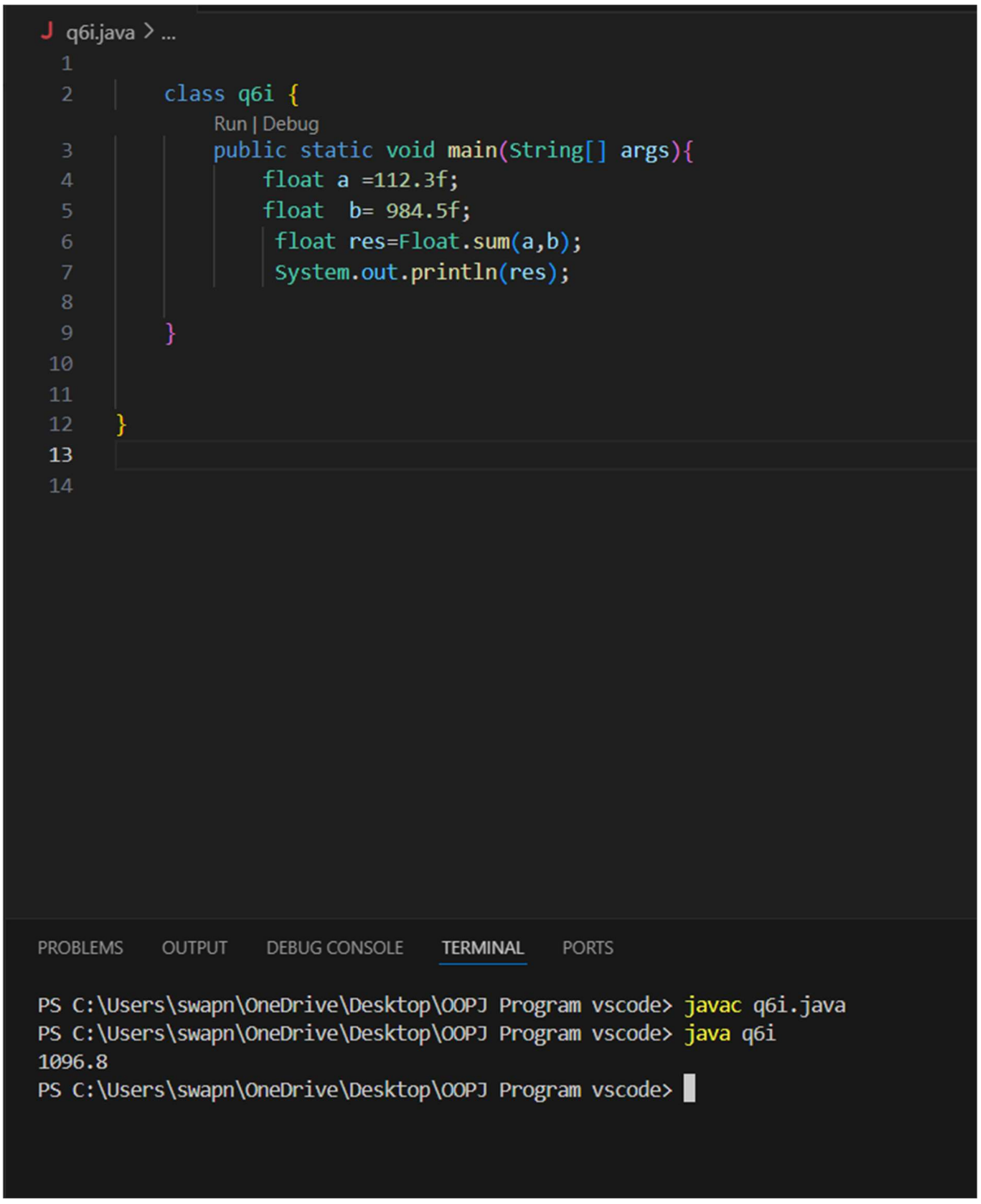
q6h.java > ...

```java
public class q6h {

    Run | Debug
    public static void main(String[] args){
        String strNumber = "1020";
       float f = Float.valueOf(strNumber);
        System.out.println(f);

        }
    }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

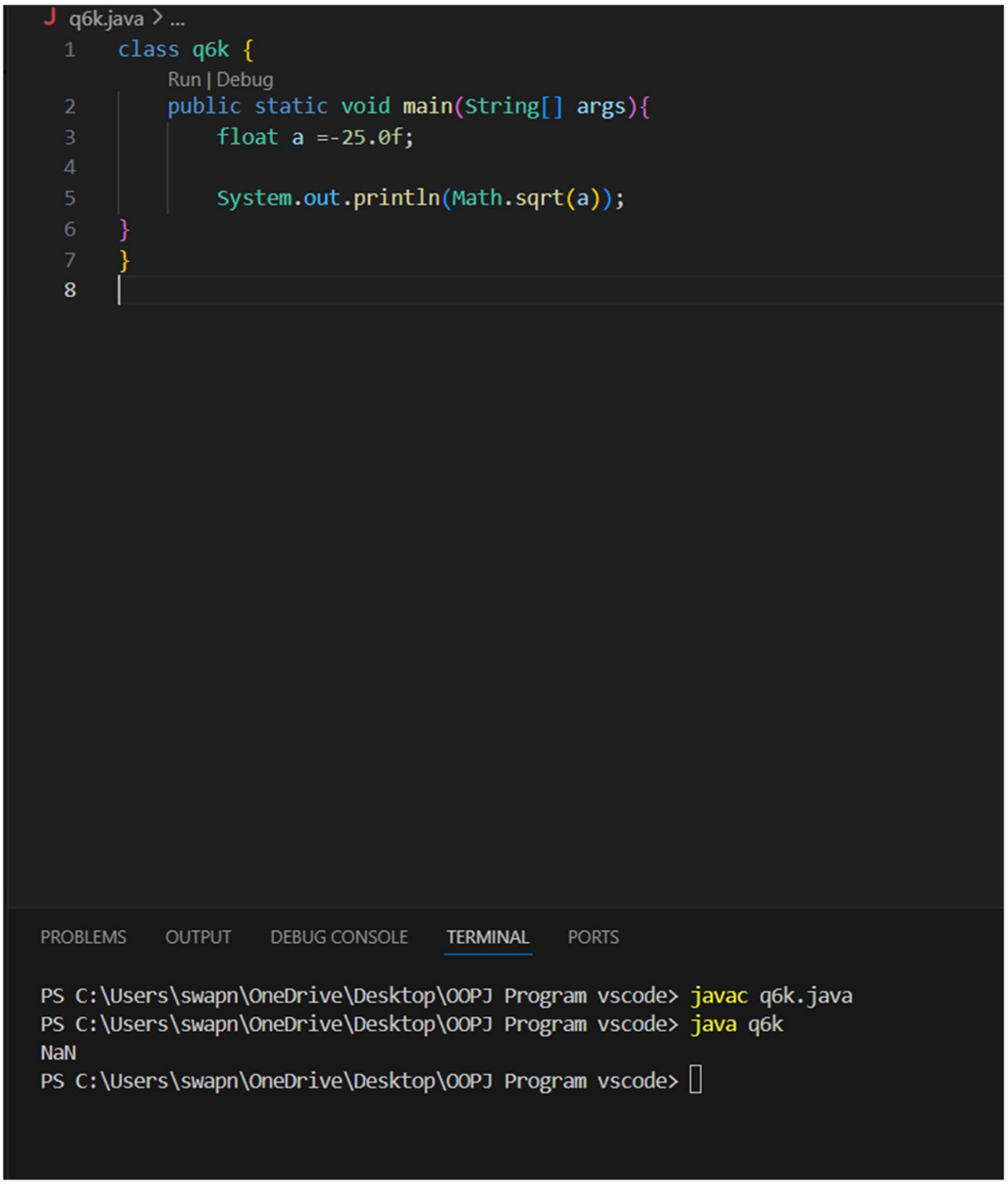
```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6h
1020.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

i. Declare two float variables with values 112.3 and 984.5, and add them using a method from the Float class.



j. Declare two float variables with values 112.2 and 556.6, and find the minimum and maximum values using the Float class.

k. Declare a float variable with the value -25.0f. Find the square root of this value.



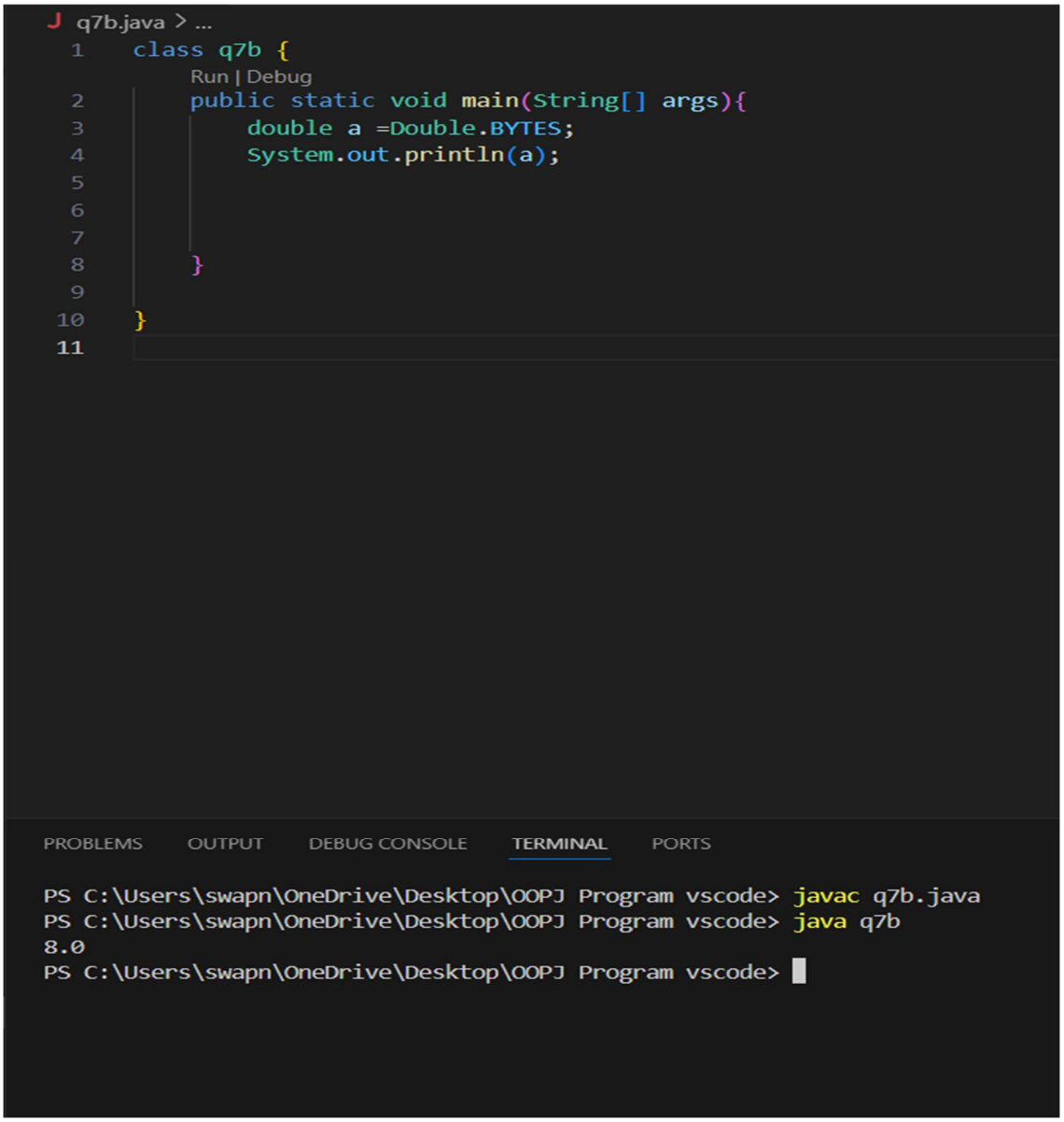
1. Declare two float variables with the same value, 0.0f, and divide them

m. Experiment with converting a float value into other primitive types or vice versa and observe the results.

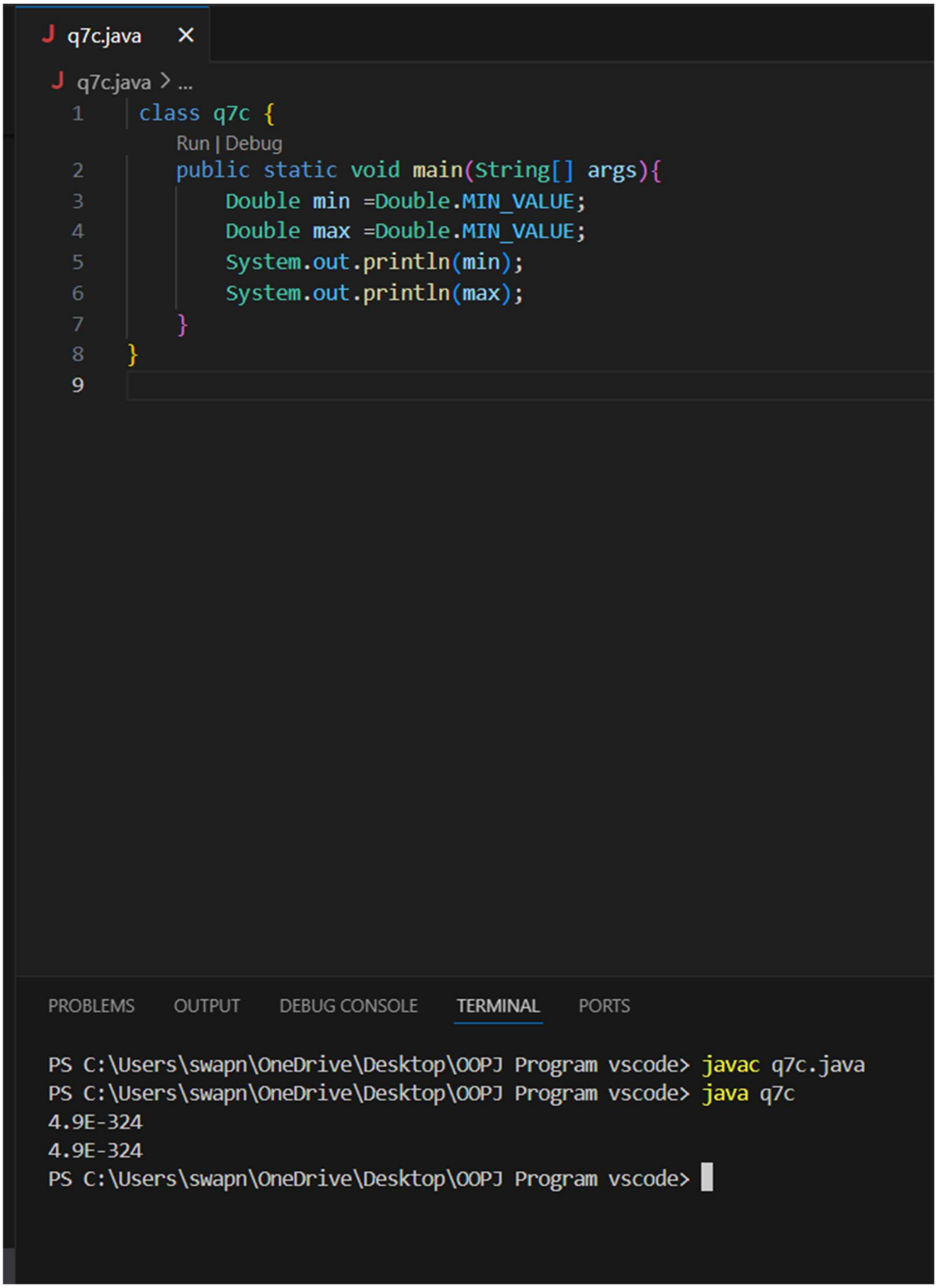
7. Working with java.lang.Double

a. Explore the Java API documentation for java.lang.Double and observe its modifiers and super types.

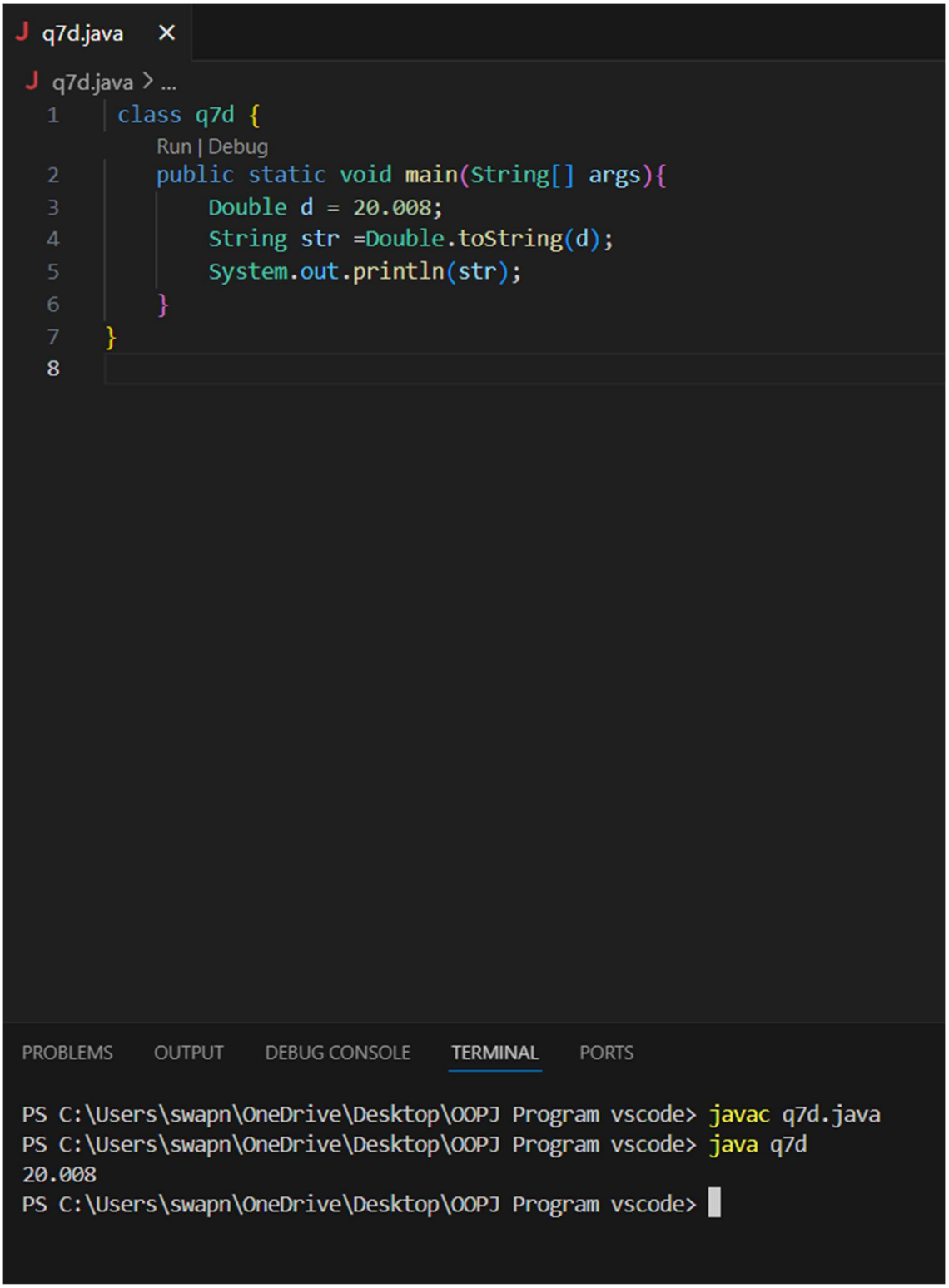
b. Write a program to test how many bytes are used to represent a double value using the BYTES field.



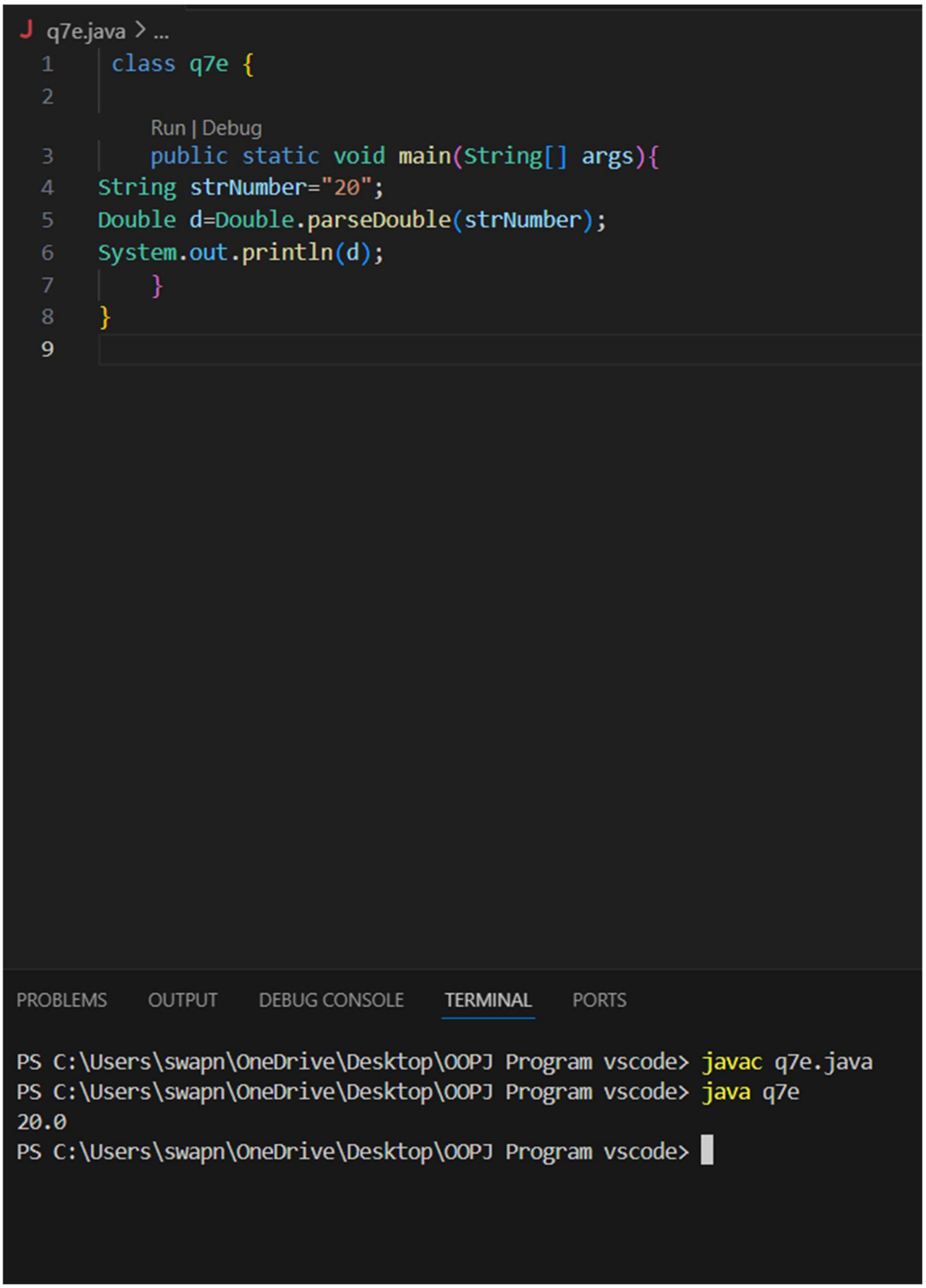
c. Write a program to find the minimum and maximum values of double using the MIN_VALUE and MAX_VALUE fields.



d. Declare a method-local variable number of type double with some value and convert it to a String using the toString method.



e. Declare a method-local variable strNumber of type String with some value and convert it to a double value using the parseDouble method.



f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a double value.

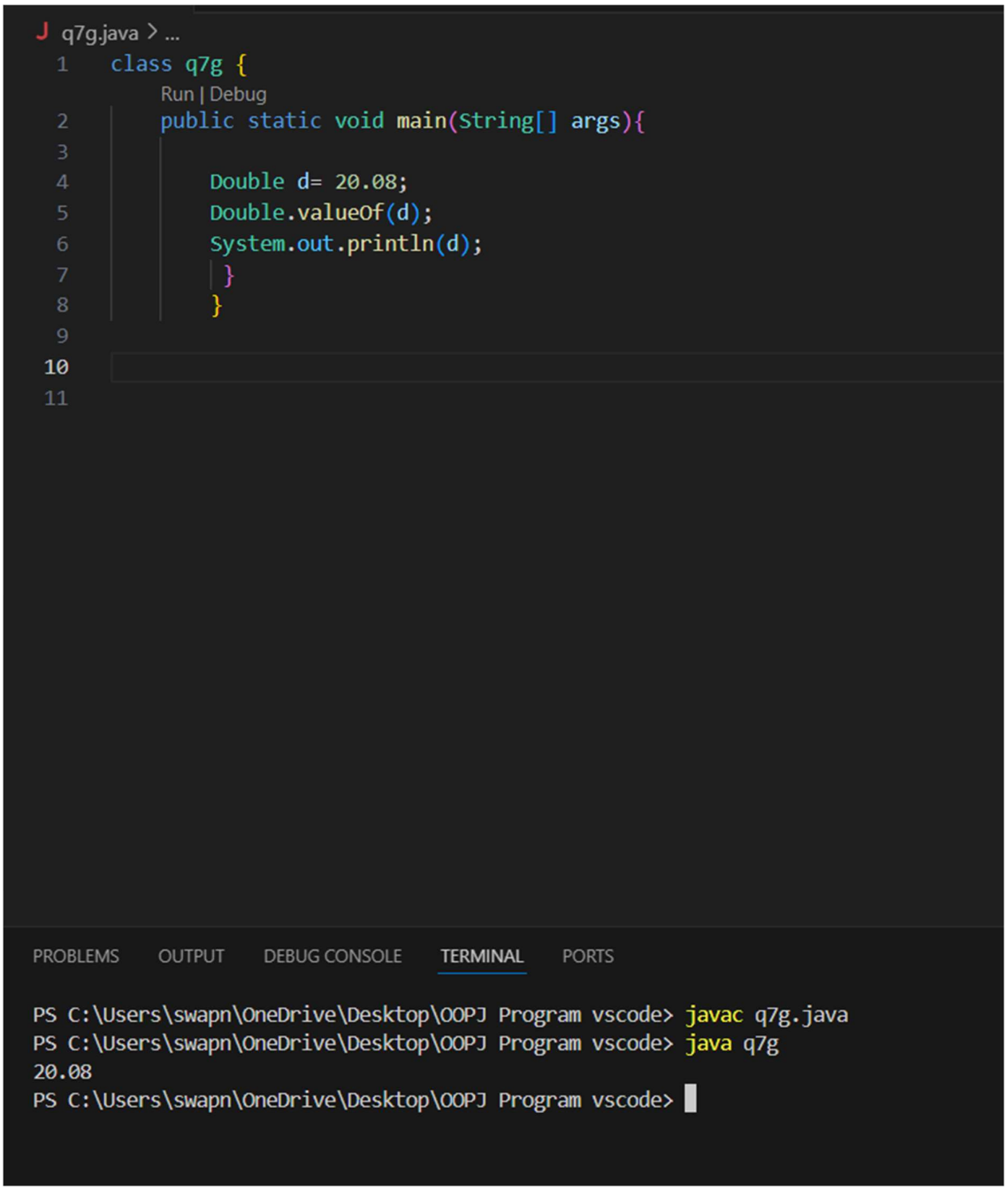
q7f.java > ...

```java
class q7f {


Run | Debug
public static void main(String[] args){
String strNumber="Ab12Cd3";
Double d=Double.parseDouble(strNumber);
System.out.println(d);
 }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

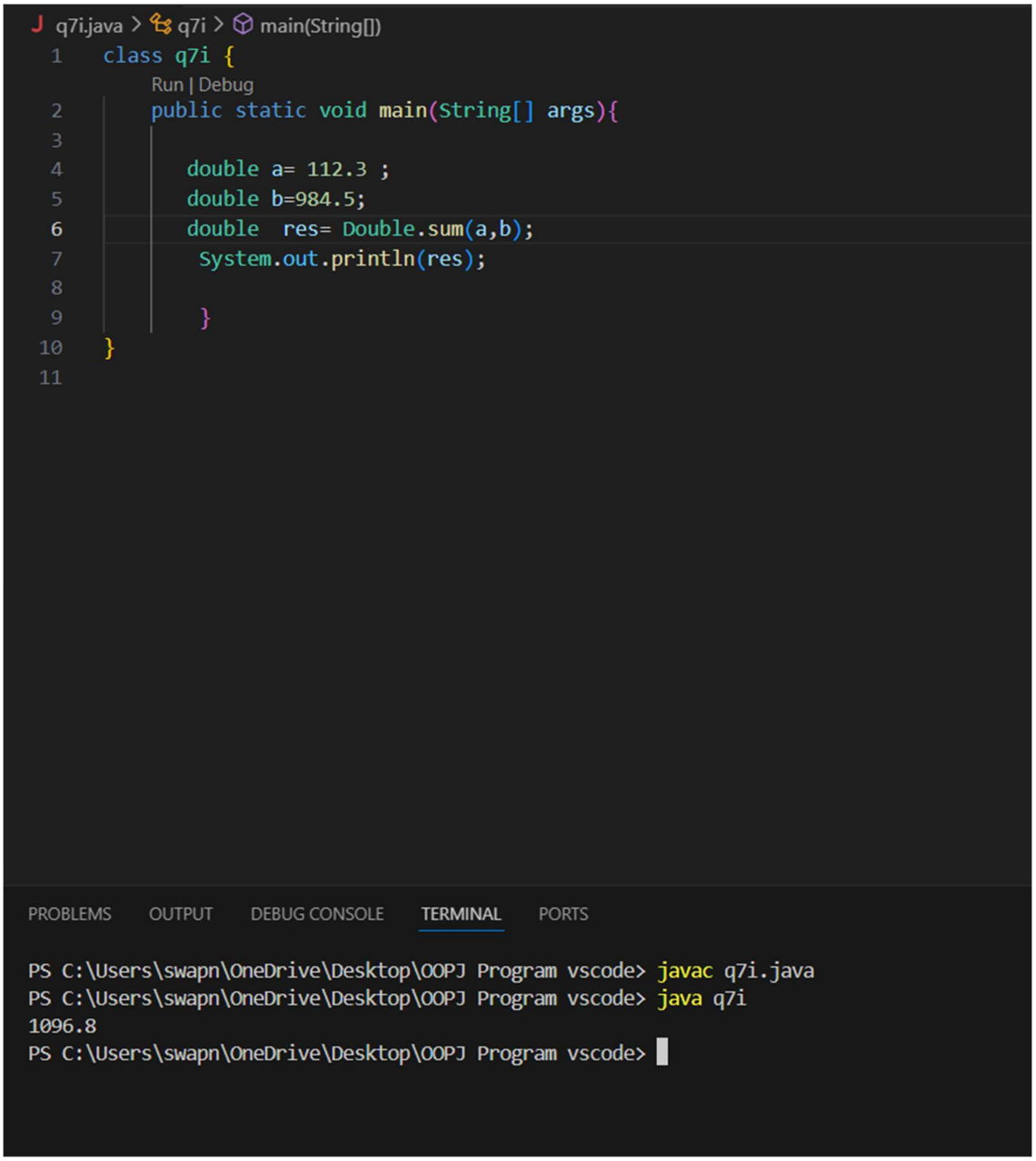
```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7f
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
        at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)
        at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
        at java.base/java.lang.Double.parseDouble(Double.java:651)
        at q7f.main(q7f.java:6)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

g. Declare a method-local variable number of type double with some value and convert it to the corresponding wrapper class using Double.valueOf().



h. Declare a method-local variable strNumber of type String with some double value and convert it to the corresponding wrapper class using Double.valueOf().

i. Declare two double variables with values 112.3 and 984.5, and add them using a method from the Double class.



j. Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class.

q7j.java > ...

```java
class q7j {
    Run | Debug
    public static void main(String[] args){

        double a= 112.2 ;
        double b=556.6;
        double min=Double.min(a,b);
        double max=Double.max(a,b);
        System.out.println("Minimum: "    + min);
         System.out.println("Maximum: " +max);


}

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7j
Minimum: 112.2
Maximum: 556.6
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

k. Declare a double variable with the value -25.0. Find the square root of this value.

q7k.java > ...

```java
class q7k {
    Run | Debug
    public static void main(String[] args){


        Double a= -25.0;

        System.out.println(Math.sqrt(a));



    }

}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7k.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7k
NaN
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

l. Declare two double variables with the same value, 0.0, and divide them.

q7l.java > ...

```java
class q7l {
    Run | Debug
    public static void main(String[] args){


        Double a= 0.0;
        Double b=0.0;

        System.out.println(a/b);


}

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7l.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7l
NaN
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

m. Experiment with converting a double value into other primitive types or vice versa and observe the results.

q7m.java > ...

```java
class q7m {
    public static void main(String[] args){
        double a= 20.8;
        System.out.println((float)a);
        System.out.println((int)a);
        System.out.println((long)a);
        System.out.println((short)a);

        int d=(int)a;
            short s =(short)a;
            System.out.println(d);
        System.out.println(s);
    }

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7m.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7m
20.8
20
20
20
20
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

8. Conversion between Primitive Types and Strings

Initialize a variable of each primitive type with a user-defined value and convert it into String:

First, use the toString method of the corresponding wrapper class. (e.g., Integer.toString()).

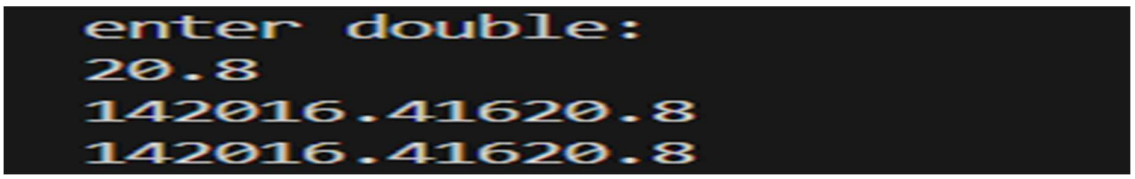
Then, use the valueOf method of the String class. (e.g., String.valueOf()).

Q8.java > ...

```java

import java.util.*;
  class Q8{
    Run | Debug
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.println(x:"enter byte:  ");
        byte b=sc.nextByte();
        System.out.println(x:"enter short:  ");
        short s=sc.nextShort();
        System.out.println(x:"enter integer:  ");
        int i=sc.nextInt();
        System.out.println(x:"enter float:  ");
        float f=sc.nextFloat();
        System.out.println(x:"enter long:  ");
        long l=sc.nextLong();
        System.out.println(x:"enter double:  ");
        double d=sc.nextDouble();

       String s1=Byte.toString(b);
       String s2=Short.toString(s);
       String s3=Integer.toString(i);
       String s4=Float.toString(f);
       String s5=Long.toString(l);
       String s6=Double.toString(d);

    System.out.println( s1 +"" + s2 + ""  + s3 + "" + s4 + "" + s5 + ""  + s6 );

```

```
   String a1=String.valueOf(b);
   String  a2=String.valueOf(s);
   String  a3=String.valueOf(i);
   String  a4=String.valueOf(f);
   String  a5=String.valueOf(l);
  String  a6=String.valueOf(d);
  System.out.println( a1 + "" + a2+ "" + a3 + "" + a4+ "" + a5+ ""  + a6 );
  sc.close();
   }
}
```

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac Q8.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java Q8
enter byte:
1
enter short:
4
enter integer:
20
enter float:
16.4
enter long:
16
```



9. Default Values of Primitive Types

Declare variables of each primitive type as fields of a class and check their default values.

Q9.java > ...

```java
class Q9 {
    short s;
    int a;
    float f;
    double d;
    long l;
    Run | Debug
    public static void main (String[] args){
        Q9 obj=  new Q9();
        System.out.println(obj.s);
        System.out.println(obj.a);
        System.out.println(obj.f);
        System.out.println(obj.d);
        System.out.println(obj.l);

    }

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac Q9.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java Q9
0
0
0.0
0.0
0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

10. Arithmetic Operations with Command Line Input

Write a program that accepts two integers and an arithmetic operator (+, -, *, /) from the command line. Perform the specified arithmetic operation based on the operator provided.

```
class Q10{
   public static void main(String args[])
   {
        if (args.length != 3) {
        System.out.println("Usage: java ArithmeticOperations <num1> <operator> <num2>");
        return;
      }

      try {
        int num1 = Integer.parseInt(args[0]);
        String operator = args[1];
        int num2 = Integer.parseInt(args[2]);
        int result;

        switch (operator) {
          case "+":
            result = num1 + num2;
            break;
          case "-":
            result = num1 - num2;
            break;
          case "*":
            result = num1 * num2;
```

```
                break;
            case "/":
                if (num2 == 0) {
                    System.out.println("Error: Division by zero is not allowed.");
                    return;
                }
                result = num1 / num2;
                break;
            default:
                System.out.println("Error: Invalid operator. Use +, -, *, or /.");
                return;
        }

        System.out.println("Result: " + result);
    } catch (NumberFormatException e) {
        System.out.println("Error: Please provide valid integers.");
    }
  }
}
```

```
PS D:\C-DAC\OOPJ\Day 2\Programs> javac Q10.java
PS D:\C-DAC\OOPJ\Day 2\Programs> java Q10
Usage: java ArithmeticOperations <num1> <operator> <num2>
PS D:\C-DAC\OOPJ\Day 2\Programs> 10 * 20
200
PS D:\C-DAC\OOPJ\Day 2\Programs>
```