ASSIGNMENT NO.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. (Hint: Use `Boolean.toString(Boolean)`).

Assignment-2 > Assignment-2(Next) > BooleanExample.java

```java
public class BooleanExample {
    public static void main(String[] args) {
        boolean status = true;

        String statusStr = Boolean.toString(status);

        System.out.println("Boolean to String: " + statusStr);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
Boolean to String: true
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

c. Declare a method-local variable `strStatus` of type `String` with the value `"true"` and convert it to a `boolean` using the `parseBoolean` method. (Hint: Use `Boolean.parseBoolean(String)`).

Assignment-2 > Assignment-2(Next) > BooleanExample.java

```java
public class BooleanExample {
    public static void main(String[] args) {
        String strStatus = "true";
        boolean status = Boolean.parseBoolean(strStatus);
        System.out.println("String to Boolean: " + status);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
String to Boolean: true
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

d. Declare a method-local variable `strStatus` of type `String` with the value `"1"` or `"0"` and attempt to convert it to a `boolean`. (Hint: `parseBoolean` method will not work as expected with `"1"` or `"0"`).

Assignment-2 > Assignment-2(Next) > BooleanExample.java

```java
public class BooleanExample {
    public static void main(String[] args) {
        String strStatus = "1";
        boolean status = Boolean.parseBoolean(strStatus);
        System.out.println("String '1' to Boolean: " + status);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
String to Boolean: true
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
String '1' to Boolean: false
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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()`. (Hint: Use `Boolean.valueOf(boolean)`).

Assignment-2 > Assignment-2(Next) > BooleanExample.java

```java
public class BooleanExample {
    public static void main(String[] args) {
        boolean status = true;
        Boolean booleanWrapper = Boolean.valueOf(status);
        System.out.println("Primitive boolean to Boolean object: " + booleanWrapper);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
Primitive boolean to Boolean object: true
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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()`. (Hint: Use `Boolean.valueOf(String)`).

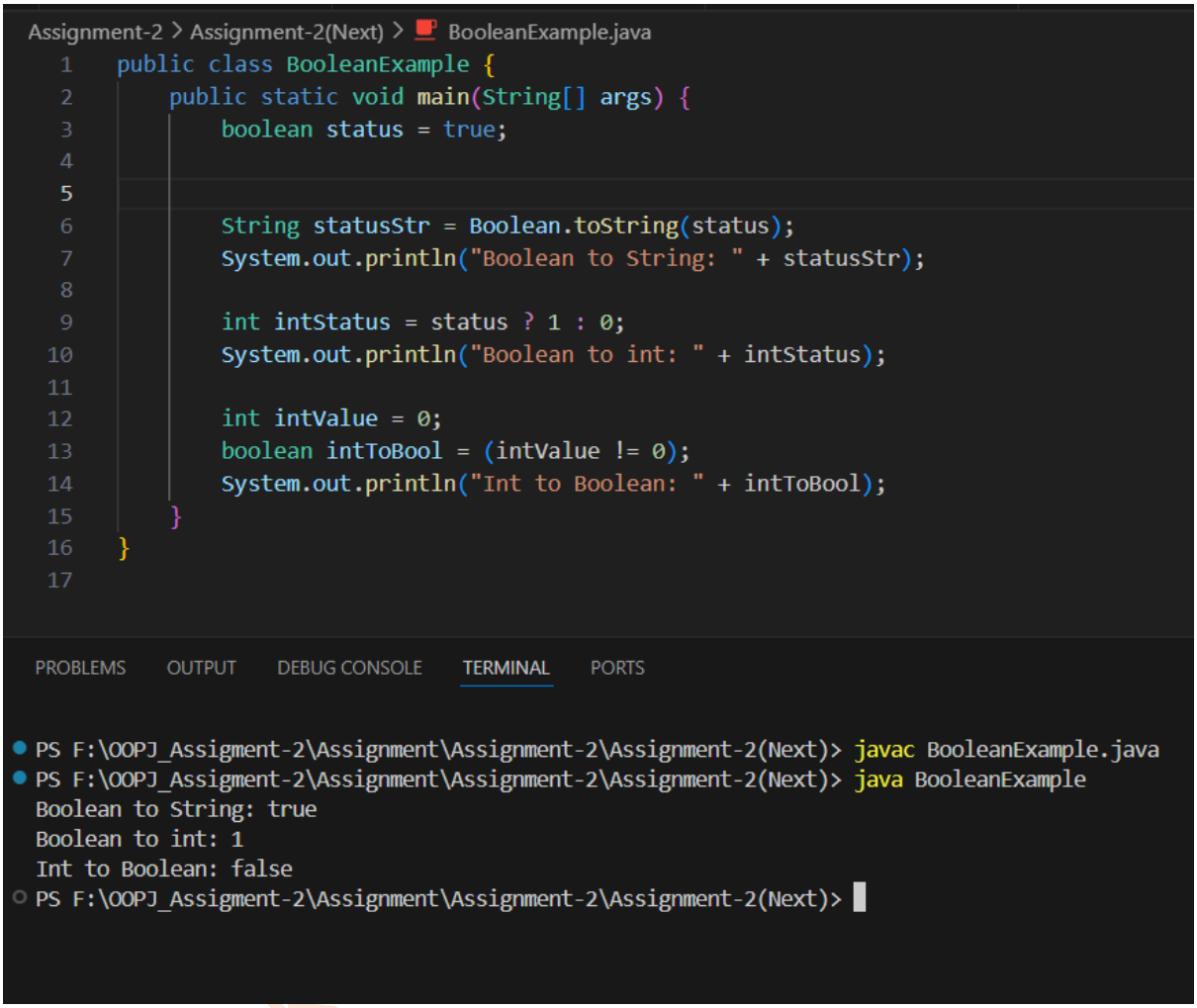
Assignment-2 > Assignment-2(Next) > BooleanExample.java

```java
public class BooleanExample {
    public static void main(String[] args) {
        String strStatus = "true";
        Boolean booleanWrapper = Boolean.valueOf(strStatus);
        System.out.println("String to Boolean object: " + booleanWrapper);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

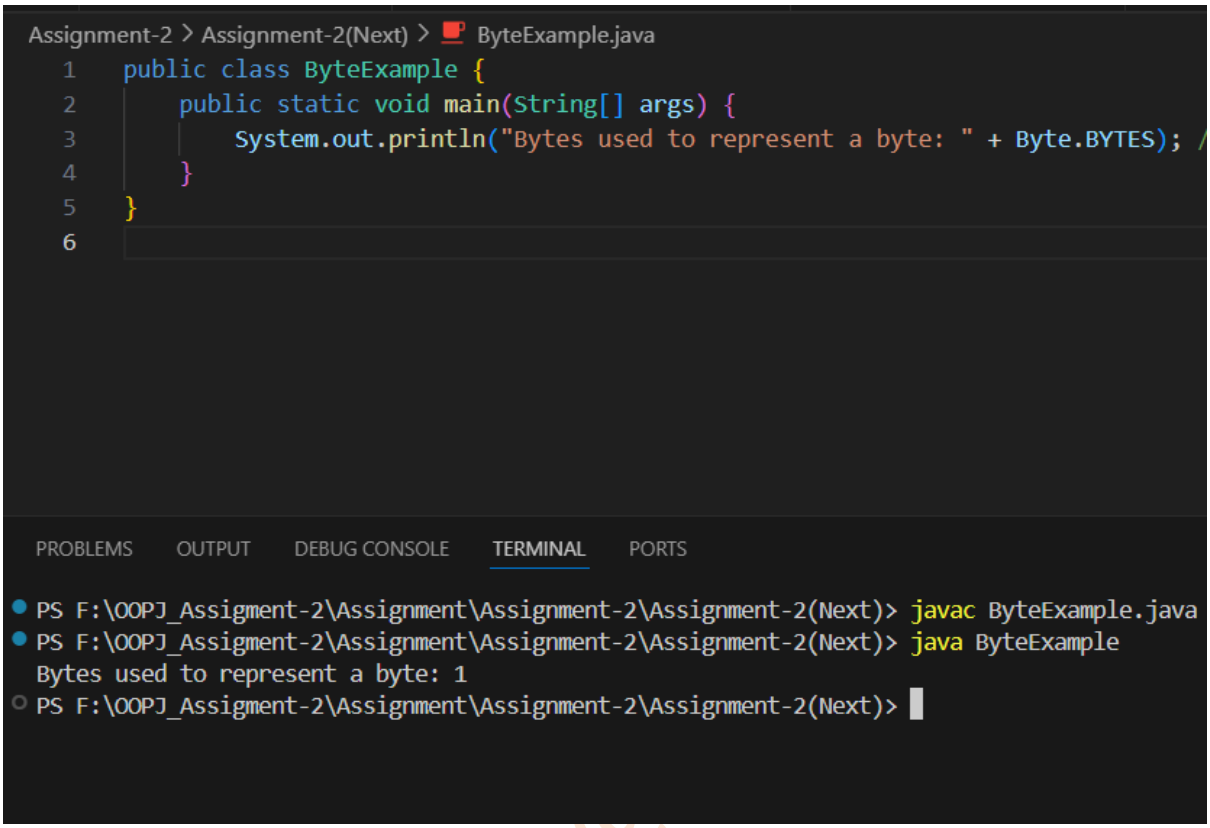
```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac BooleanExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java BooleanExample
String to Boolean object: true
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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

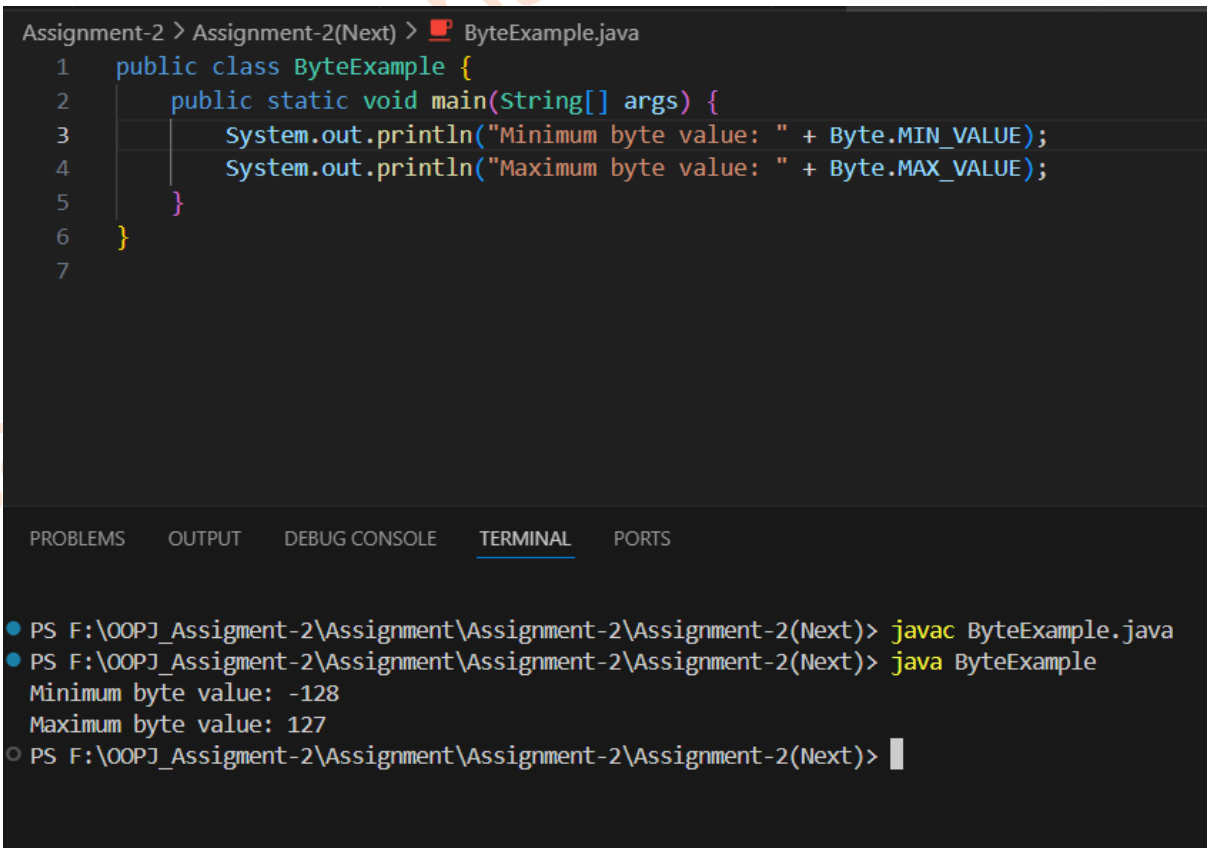


2. Working with `java.lang.Byte`

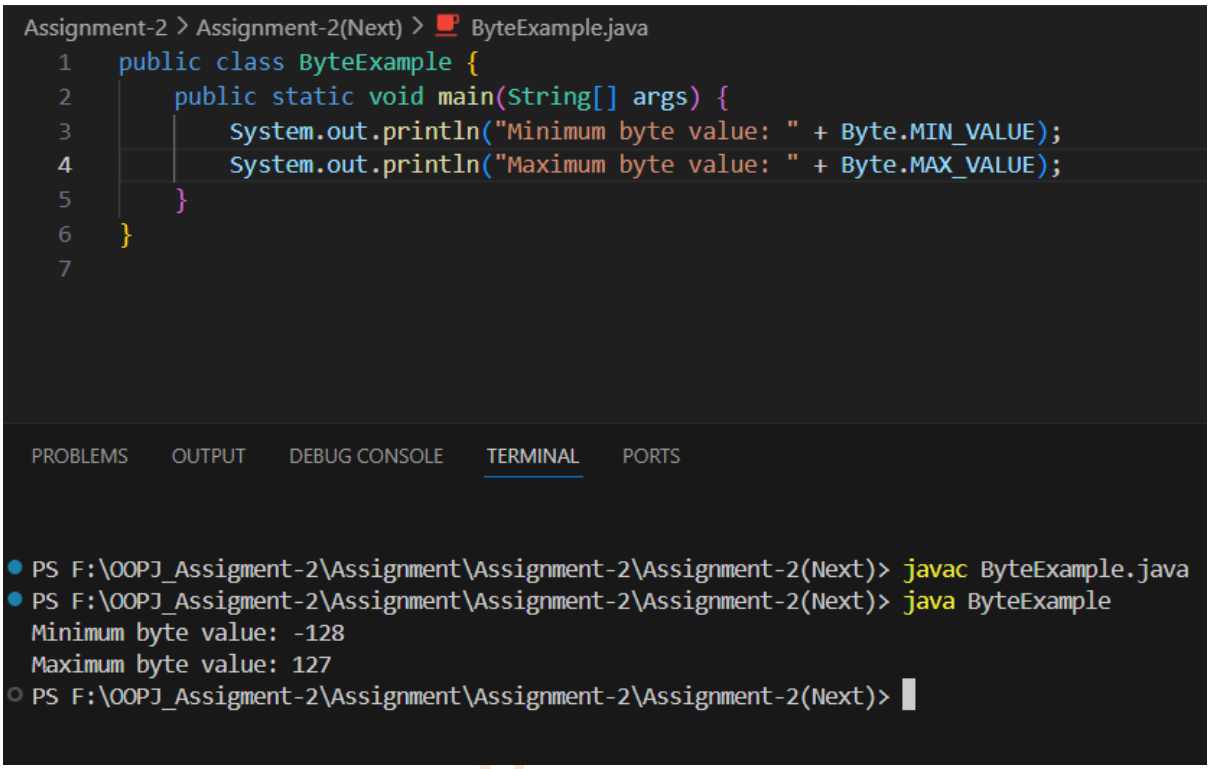
a. Explore the Java API documentation for `java.lang.Byte` and observe its modifiers and super types.



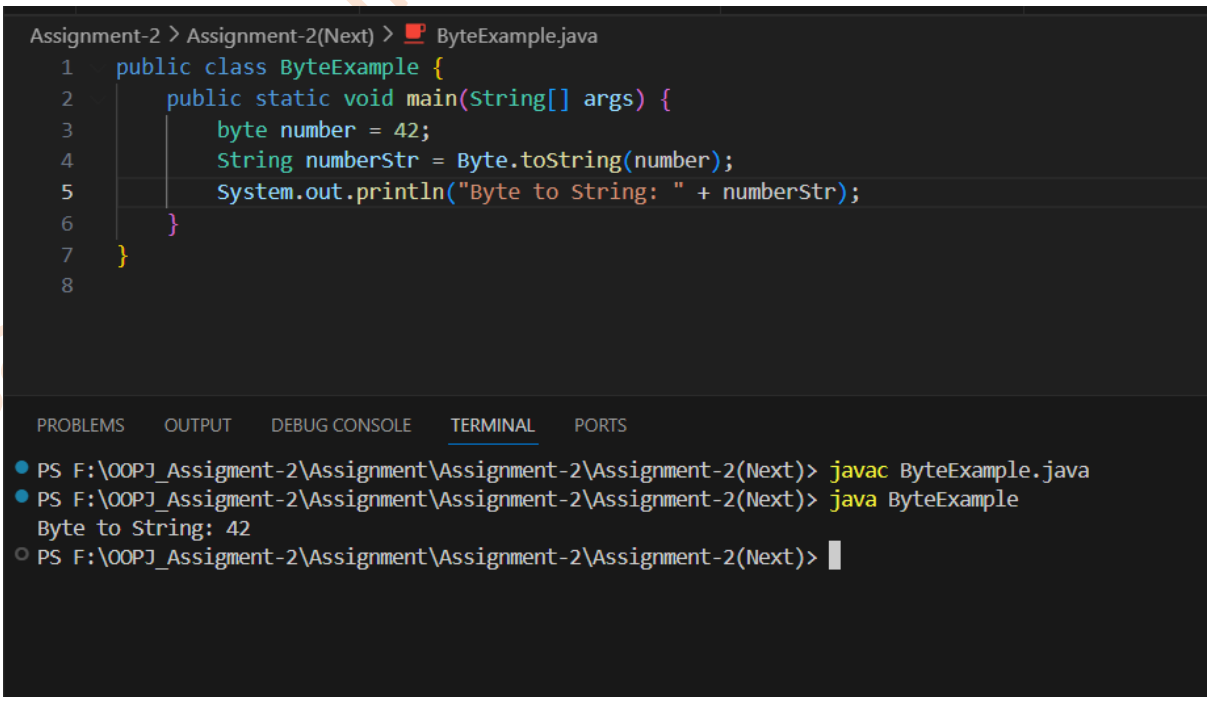
b. Write a program to test how many bytes are used to represent a `byte` value using the `BYTES` field. (Hint: Use `Byte.BYTES`).



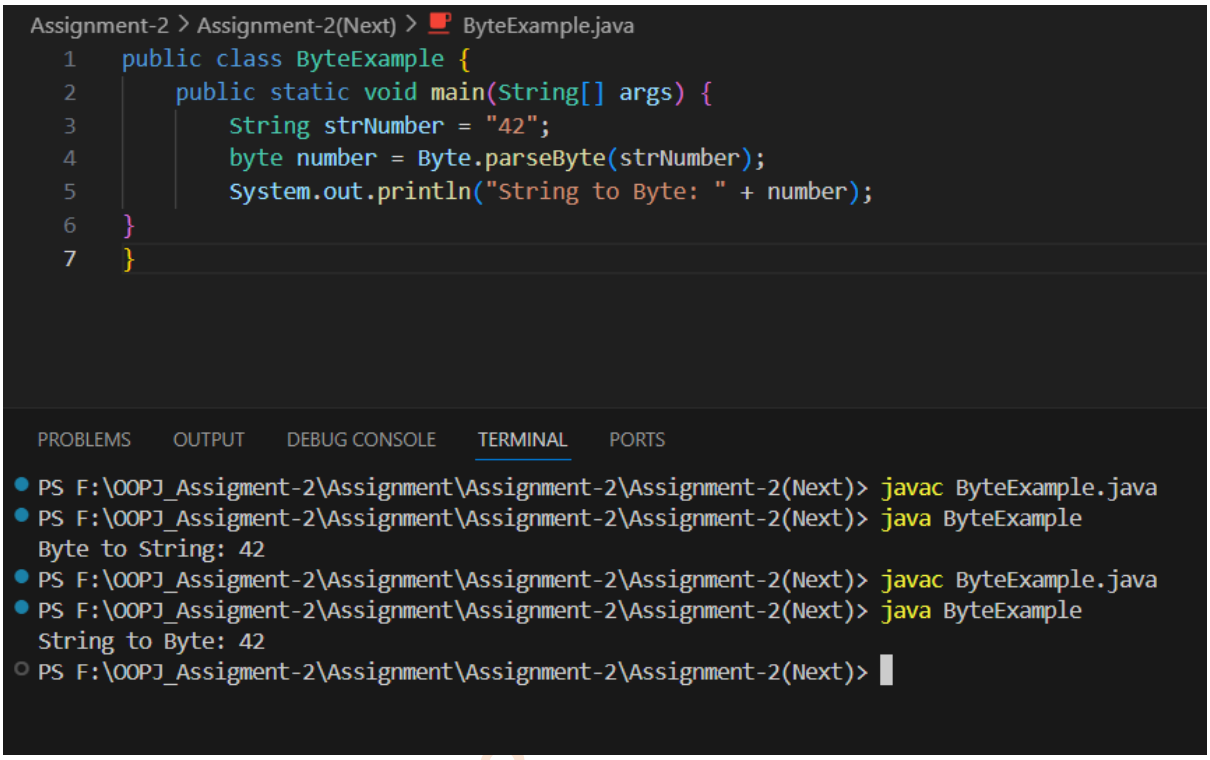
c. Write a program to find the minimum and maximum values of `byte` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Byte.MIN_VALUE` and `Byte.MAX_VALUE`).



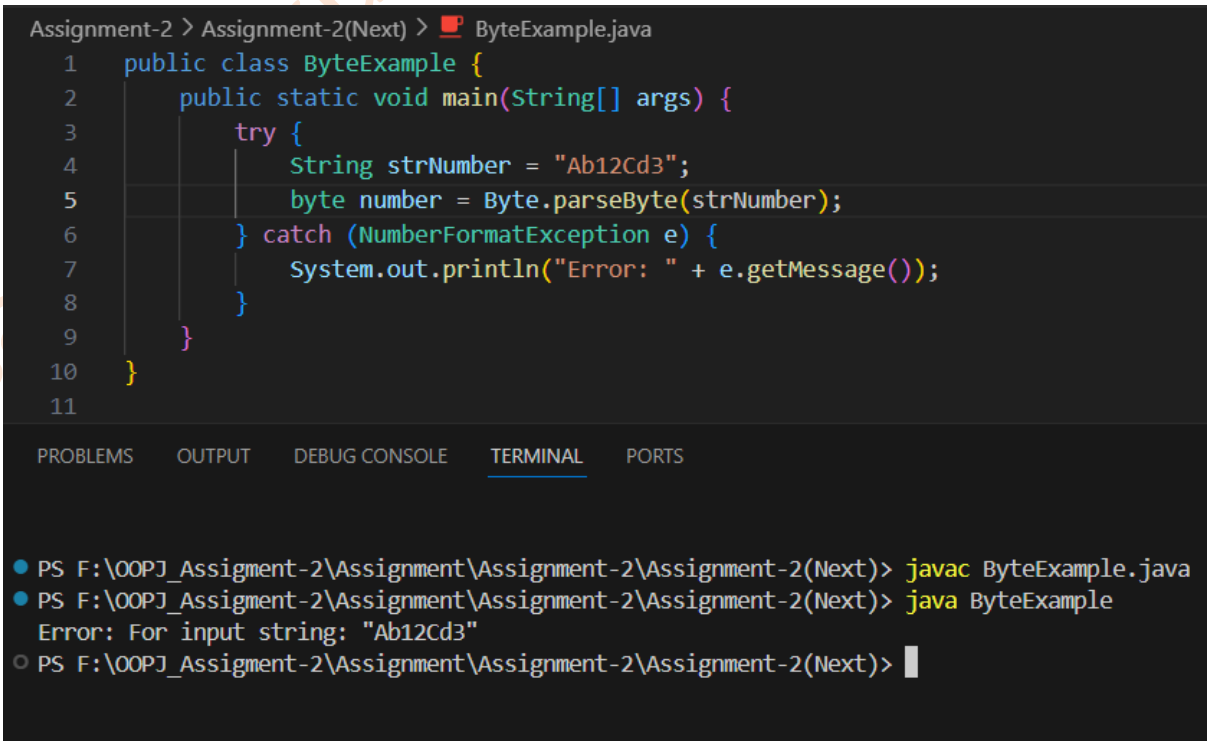
d. Declare a method-local variable `number` of type `byte` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Byte.toString(byte)`).



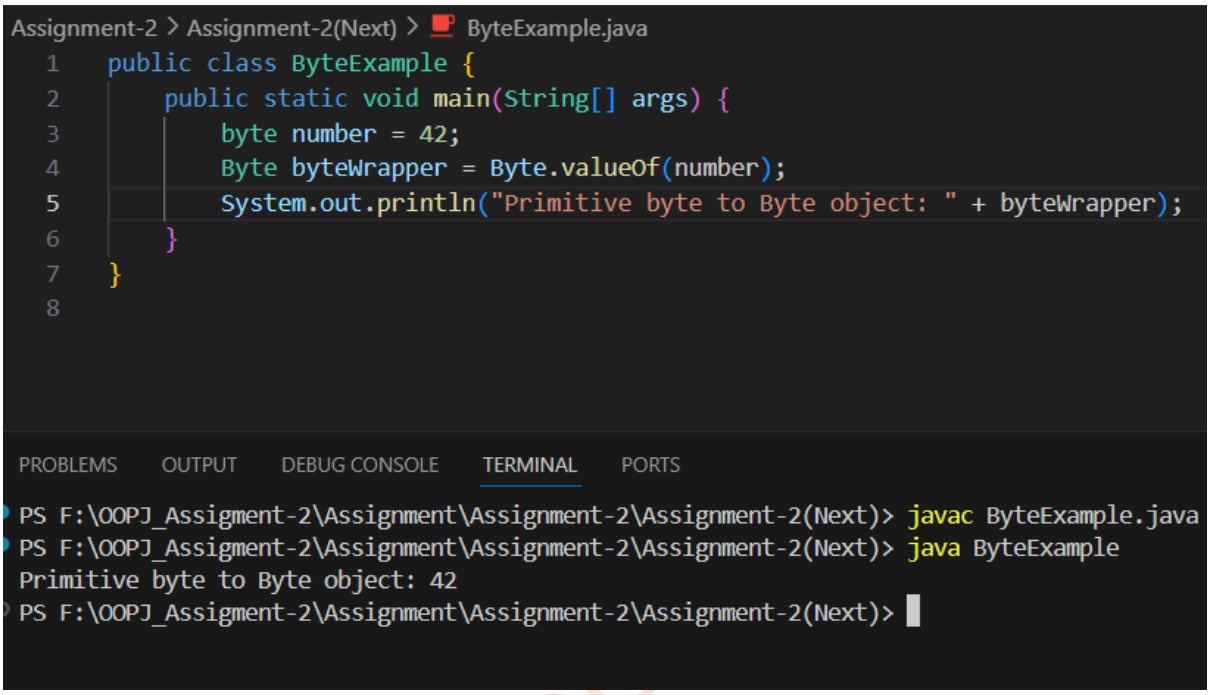
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `byte` value using the `parseByte` method. (Hint: Use `Byte.parseByte(String)`).



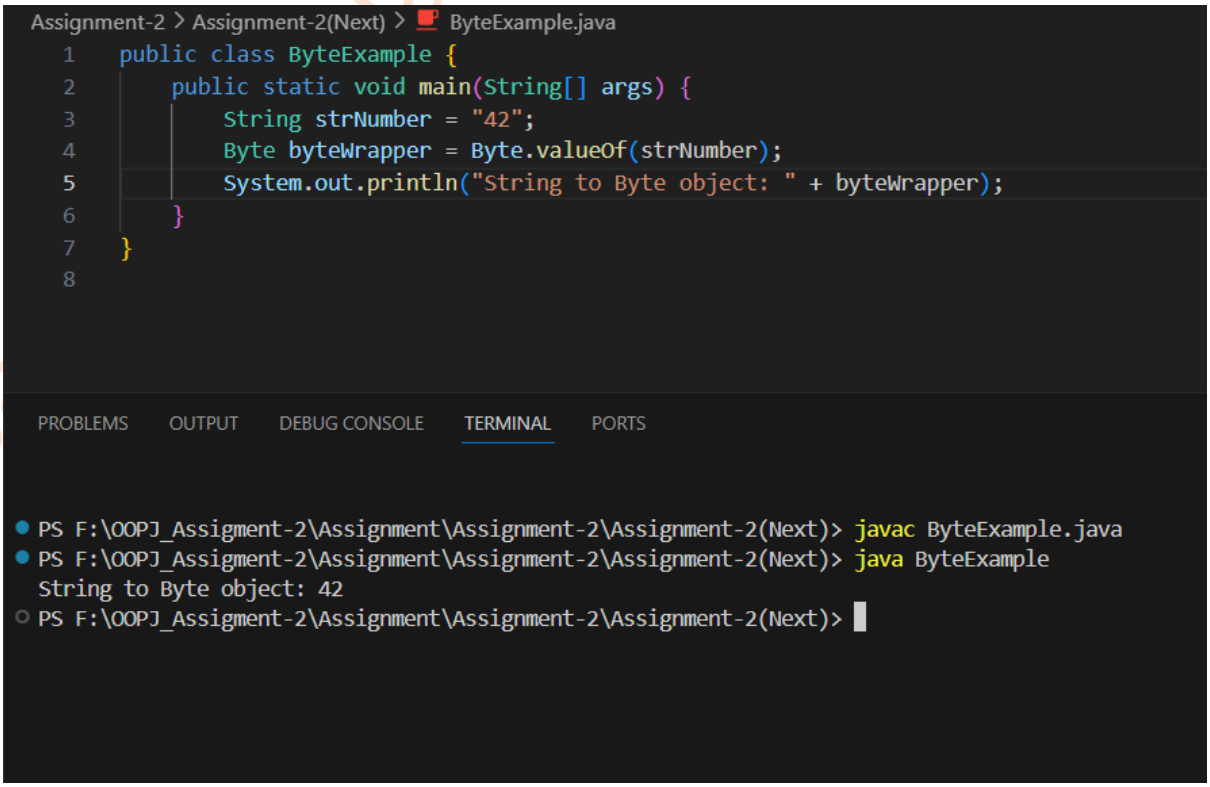
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `byte` value. (Hint: `parseByte` method will throw a `NumberFormatException`).



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



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()`. (Hint: Use `Byte.valueOf(String)`).



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

ii.

Assignment-2 > Assignment-2(Next) > ByteExample.java

```java
public class ByteExample {
    public static void main(String[] args) {
        byte number = 42;

        int intValue = number;
        System.out.println("Byte to int: " + intValue);

        int anotherInt = 100;
        byte intToByte = (byte) anotherInt;
        System.out.println("Int to byte: " + intToByte);
    }
}
```

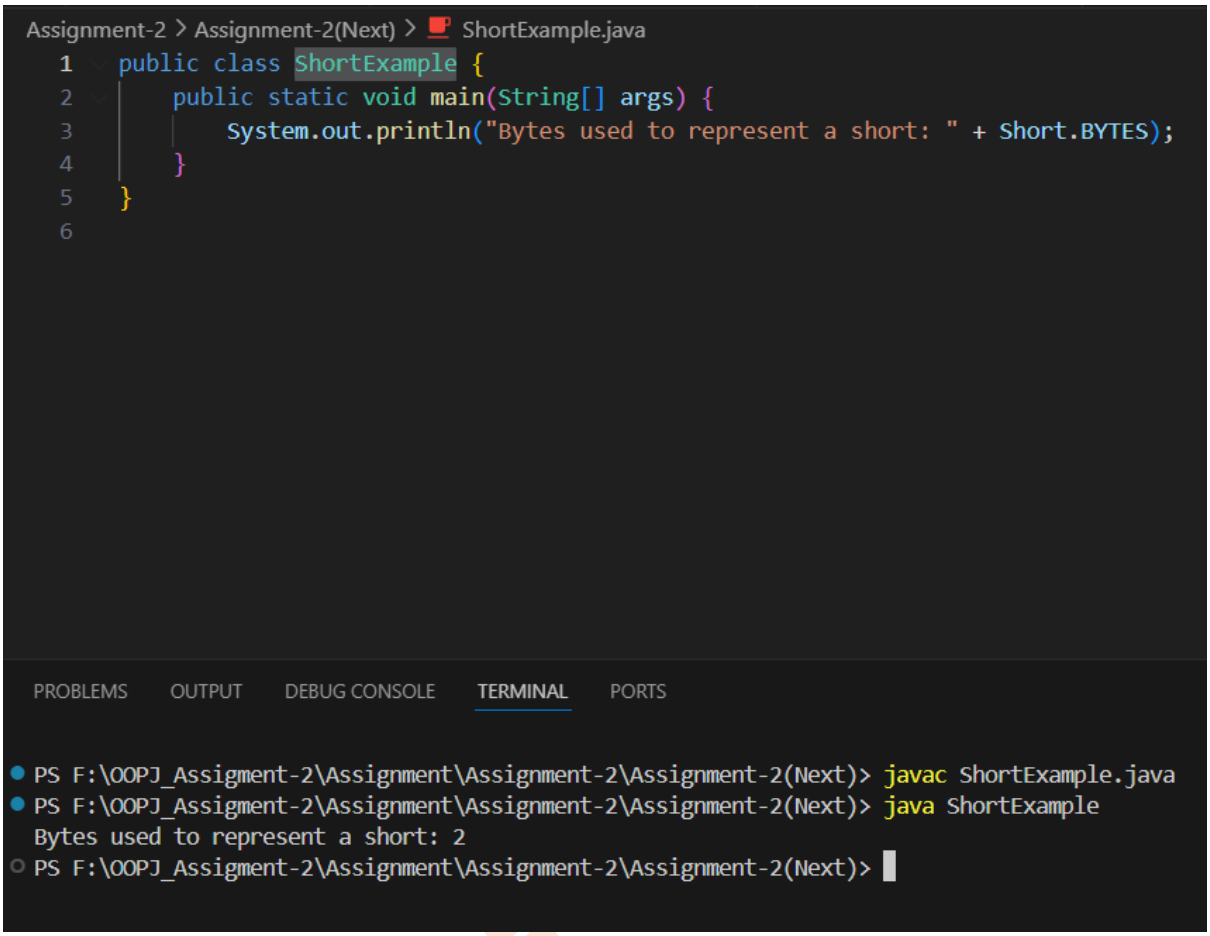
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ByteExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ByteExample
Byte to int: 42
Int to byte: 100
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

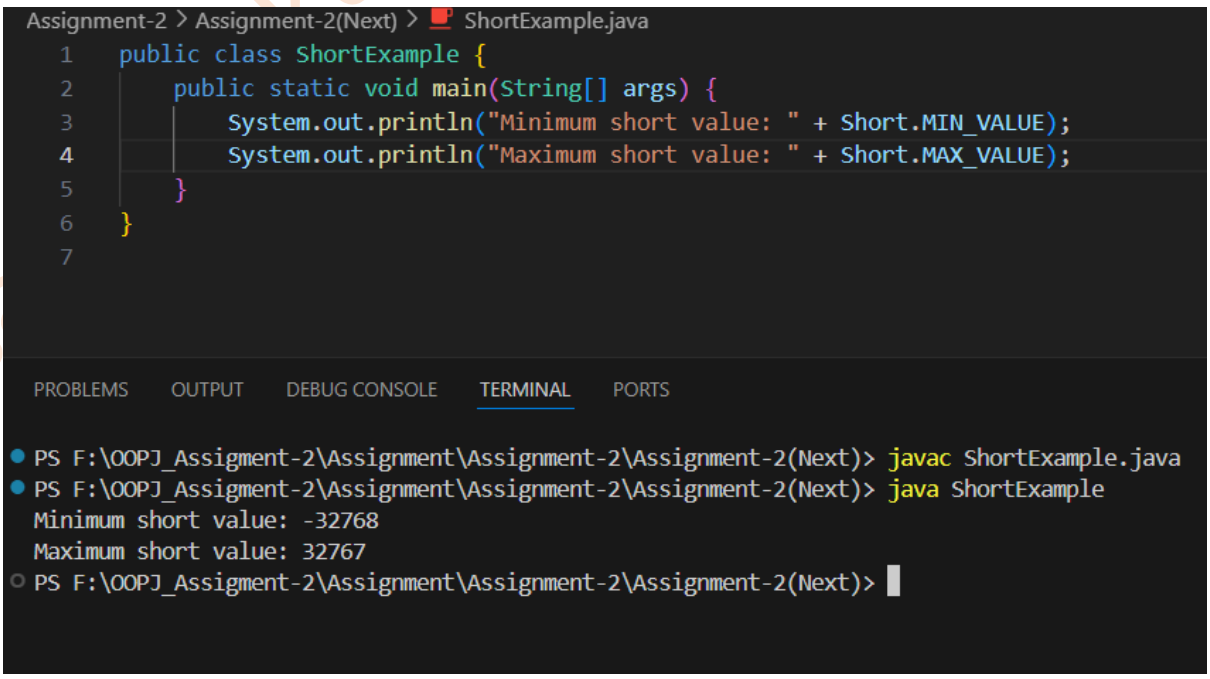
3. Working with `java.lang.Short`

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. (Hint: Use `Short.BYTES`).



c. Write a program to find the minimum and maximum values of `short` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Short.MIN_VALUE` and `Short.MAX_VALUE`).



d. Declare a method-local variable `number` of type `short` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Short.toString(short)`).

Assignment-2 > Assignment-2(Next) > ShortExample.java

```java
public class ShortExample {
    public static void main(String[] args) {
        short number = 12345;
        String numberStr = Short.toString(number);
        System.out.println("Short to String: " + numberStr);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ShortExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ShortExample
Minimum short value: -32768
Maximum short value: 32767
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ShortExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ShortExample
Short to String: 12345
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `short` value using the `parseShort` method. (Hint: Use `Short.parseShort(String)`).

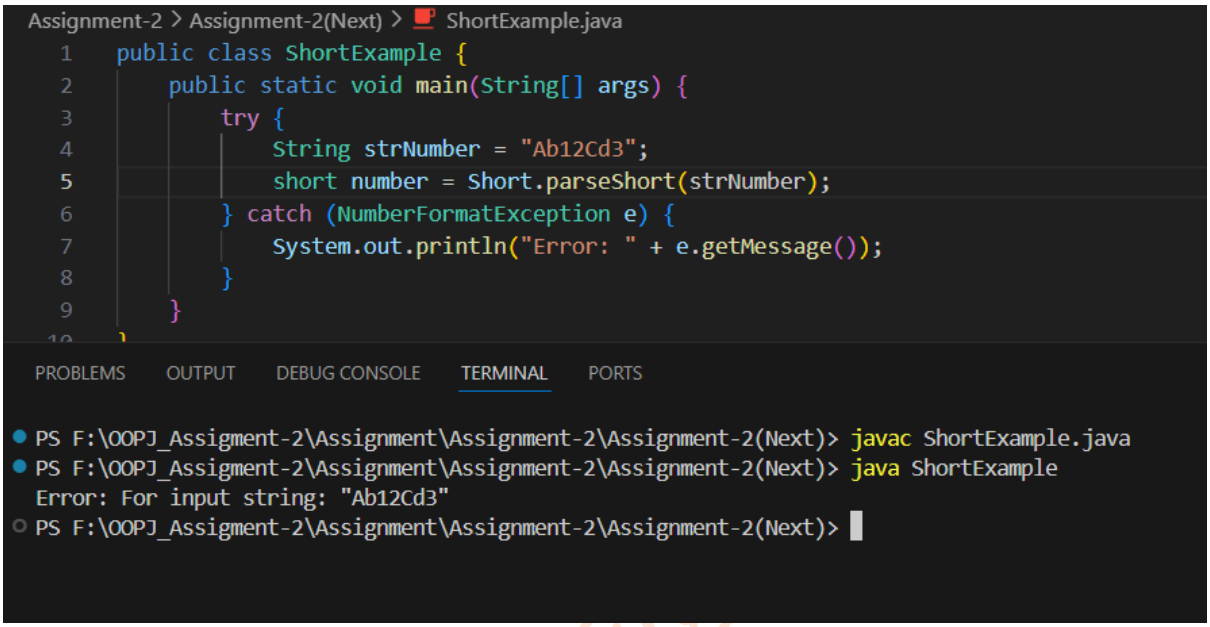
Assignment-2 > Assignment-2(Next) > ShortExample.java

```java
public class ShortExample {
    public static void main(String[] args) {
        String strNumber = "12345";
        short number = Short.parseShort(strNumber);
        System.out.println("String to Short: " + number);
    }
}
```

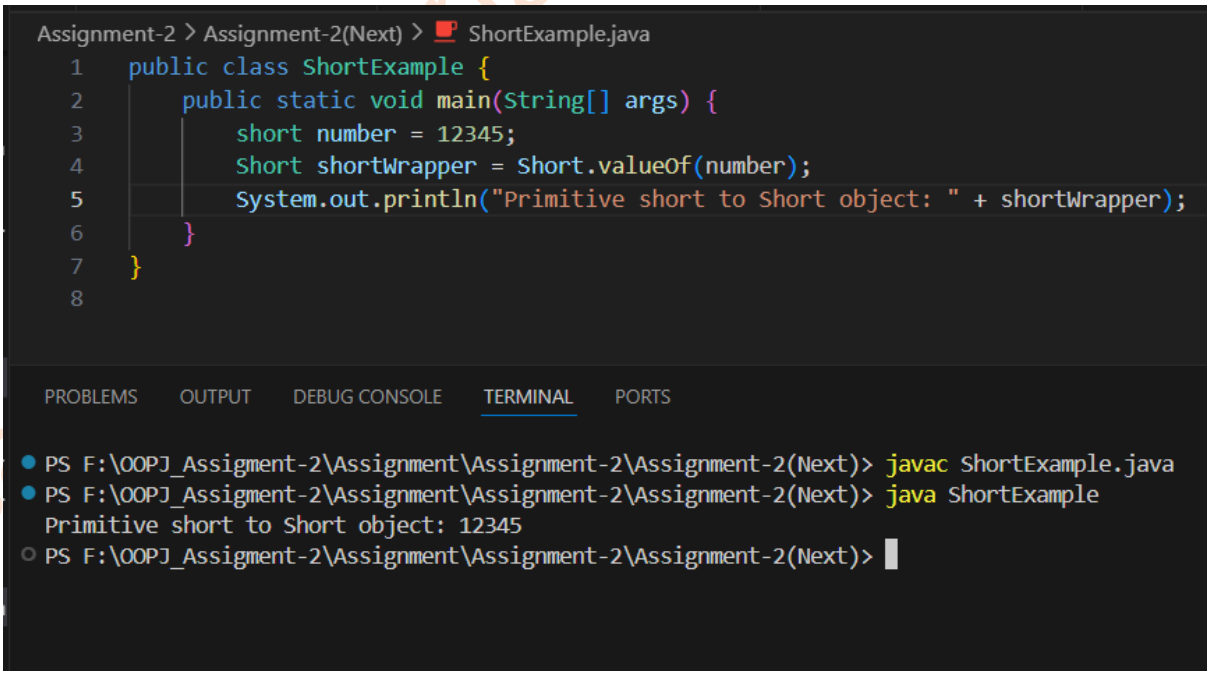
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ShortExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ShortExample
String to Short: 12345
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `short` value. (Hint: `parseShort` method will throw a `NumberFormatException`).



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



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()`. (Hint: Use `Short.valueOf(String)`).

Assignment-2 > Assignment-2(Next) > ShortExample.java

```java
public class ShortExample {
    public static void main(String[] args) {
        String strNumber = "12345";
        Short shortWrapper = Short.valueOf(strNumber);
        System.out.println("String to Short object: " + shortWrapper);
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ShortExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ShortExample
String to Short object: 12345
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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

Assignment-2 > Assignment-2(Next) > ShortExample.java

```java
public class ShortExample {
    public static void main(String[] args) {
        short number = 12345;

        int intValue = number;
        System.out.println("Short to int: " + intValue);

        int anotherInt = 32000;
        short intToShort = (short) anotherInt;
        System.out.println("Int to short: " + intToShort);
    }
}
```

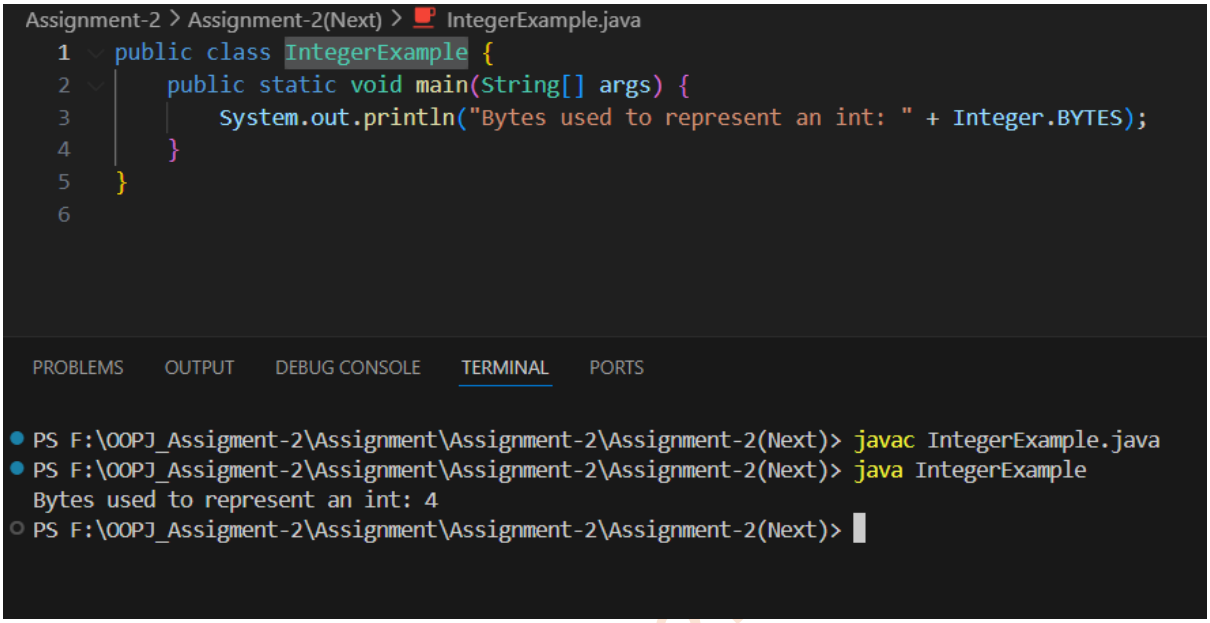
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ShortExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ShortExample
Short to int: 12345
Int to short: 32000
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

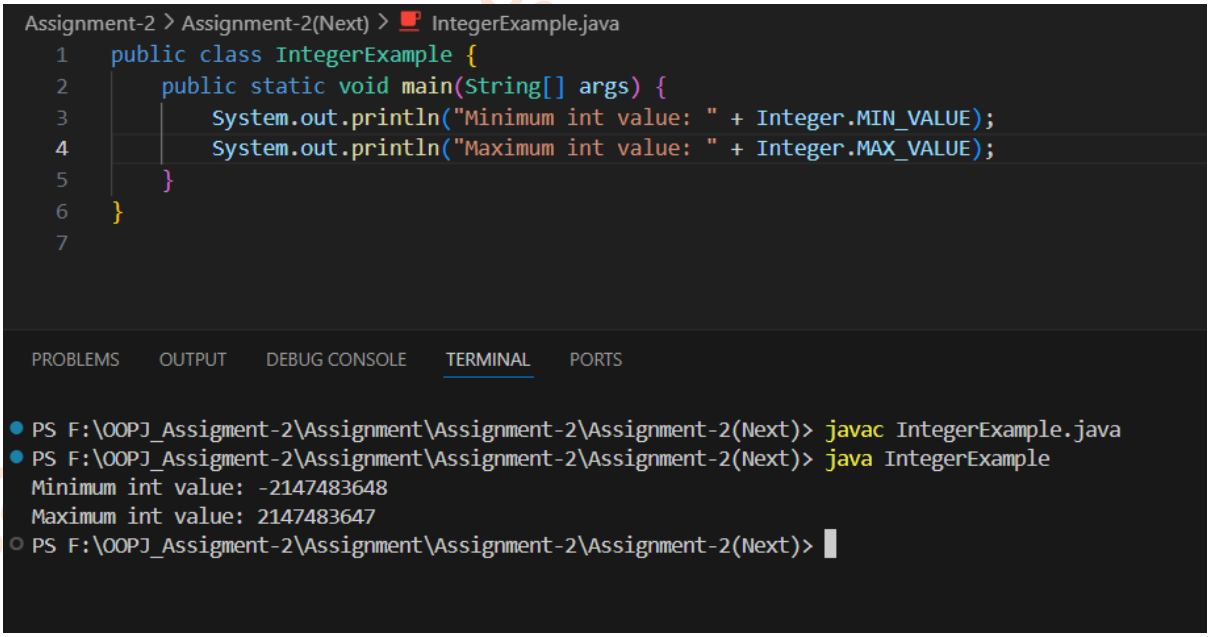
4. Working with `java.lang.Integer`

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. (Hint: Use `Integer.BYTES`).



c. Write a program to find the minimum and maximum values of `int` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Integer.MIN_VALUE` and `Integer.MAX_VALUE`).



d. Declare a method-local variable `number` of type `int` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Integer.toString(int)`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int number = 123;
        String numberStr = Integer.toString(number);
        System.out.println("Int to String: " + numberStr);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Int to String: 123
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to an `int` value using the `parseInt` method. (Hint: Use `Integer.parseInt(String)`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        String strNumber = "123";
        int number = Integer.parseInt(strNumber);
        System.out.println("String to Int: " + number);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
String to Int: 123
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to an `int` value. (Hint: `parseInt` method will throw a `NumberFormatException`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        try {
            String strNumber = "Ab12Cd3";
            int number = Integer.parseInt(strNumber);
        } catch (NumberFormatException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Error: For input string: "Ab12Cd3"
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int number = 123;
        Integer intWrapper = Integer.valueOf(number);
        System.out.println("Primitive int to Integer object: " + intWrapper);
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Primitive int to Integer object: 123
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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()`. (Hint: Use `Integer.valueOf(String)`).

ASSIGNMENT NO.2

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        String strNumber = "123";
        Integer intWrapper = Integer.valueOf(strNumber);
        System.out.println("String to Integer object: " + intWrapper);
    }
}

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
String to Integer object: 123
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

i. Declare two integer variables with values `10` and `20`, and add them using a method from the `Integer` class. (Hint: Use `Integer.sum(int, int)`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int num1 = 10;
        int num2 = 20;
        int sum = Integer.sum(num1, num2);
        System.out.println("Sum of 10 and 20: " + sum);
    }
}

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Sum of 10 and 20: 30
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

j. Declare two integer variables with values `10` and `20`, and find the minimum and maximum values using the `Integer` class. (Hint: Use `Integer.min(int, int)` and `Integer.max(int, int)`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int num1 = 10;
        int num2 = 20;
        int min = Integer.min(num1, num2);
        int max = Integer.max(num1, num2);
        System.out.println("Min: " + min + ", Max: " + max);
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Min: 10, Max: 20
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

k. Declare an integer variable with the value `7`. Convert it to binary, octal, and hexadecimal strings using methods from the `Integer` class. (Hint: Use `Integer.toBinaryString(int)`, `Integer.toOctalString(int)`, and `Integer.toHexString(int)`).

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int number = 7;
        System.out.println("Binary: " + Integer.toBinaryString(number));
        System.out.println("Octal: " + Integer.toOctalString(number));
        System.out.println("Hexadecimal: " + Integer.toHexString(number));
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Binary: 111
Octal: 7
Hexadecimal: 7
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

ASSIGNMENT NO.2

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

Assignment-2 > Assignment-2(Next) > IntegerExample.java

```java
public class IntegerExample {
    public static void main(String[] args) {
        int number = 123;

        double doubleValue = (double) number;
        System.out.println("Int to double: " + doubleValue);

        double anotherDouble = 123.45;
        int doubleToInt = (int) anotherDouble;
        System.out.println("Double to int: " + doubleToInt);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac IntegerExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java IntegerExample
Int to double: 123.0
Double to int: 123
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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 `YTES` field. (Hint: Use `Long.BYTES`).

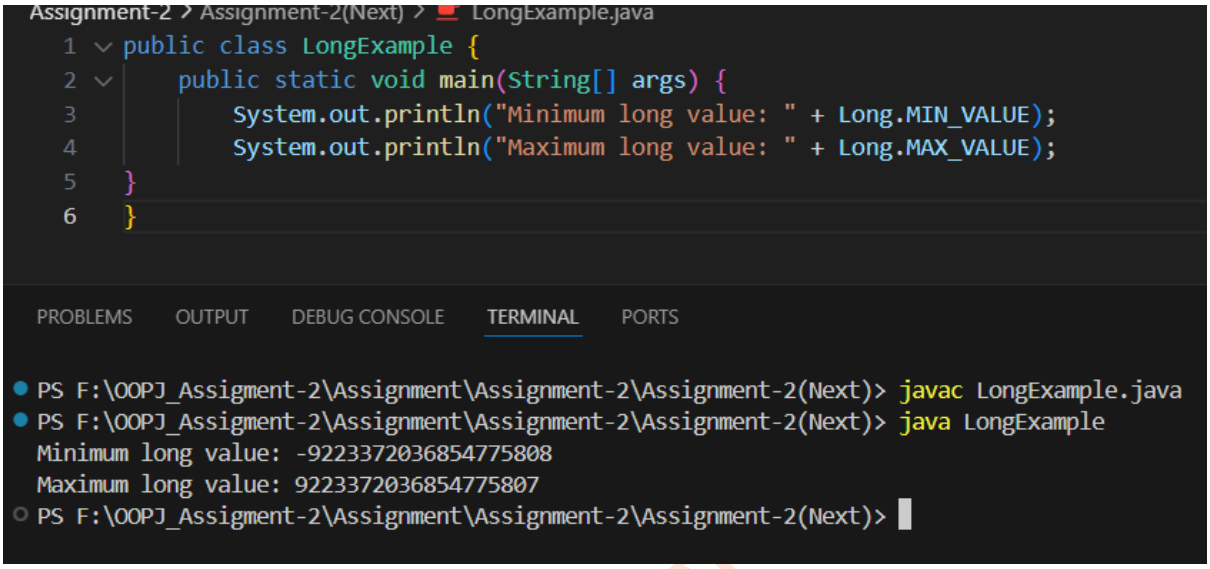
Assignment-2 > Assignment-2(Next) > LongExample.java

```java
public class LongExample {
    public static void main(String[] args) {
        System.out.println("Bytes used to represent a long: " + Long.BYTES);
    }
}
```

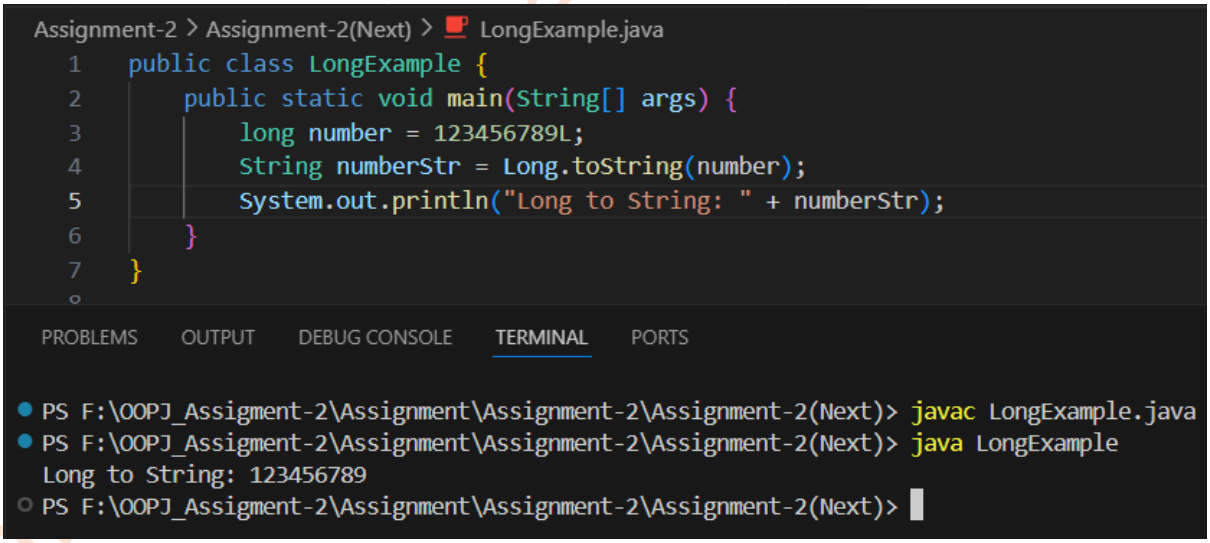
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac LongExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java LongExample
Bytes used to represent a long: 8
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

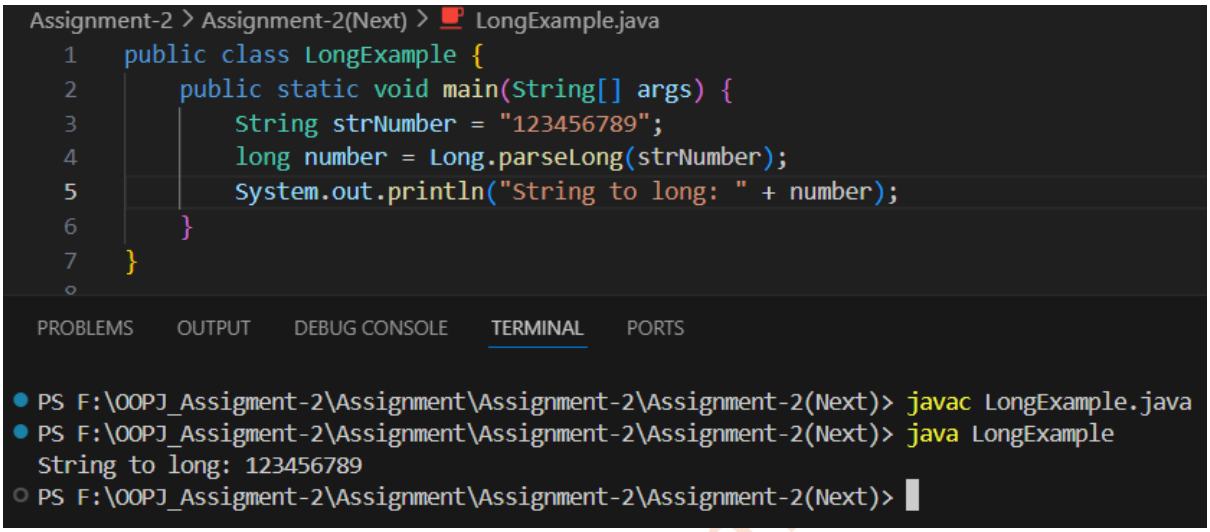
c. Write a program to find the minimum and maximum values of `long` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Long.MIN_VALUE` and `Long.MAX_VALUE`).



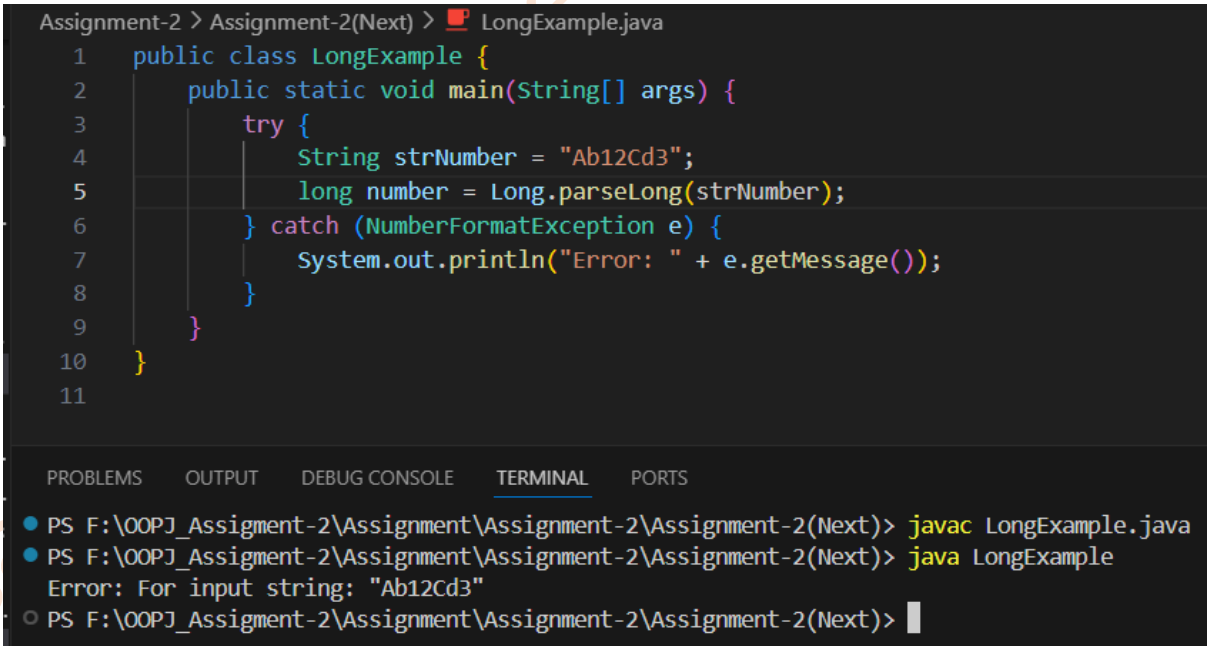
d. Declare a method-local variable `number` of type `long` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Long.toString(long)`).



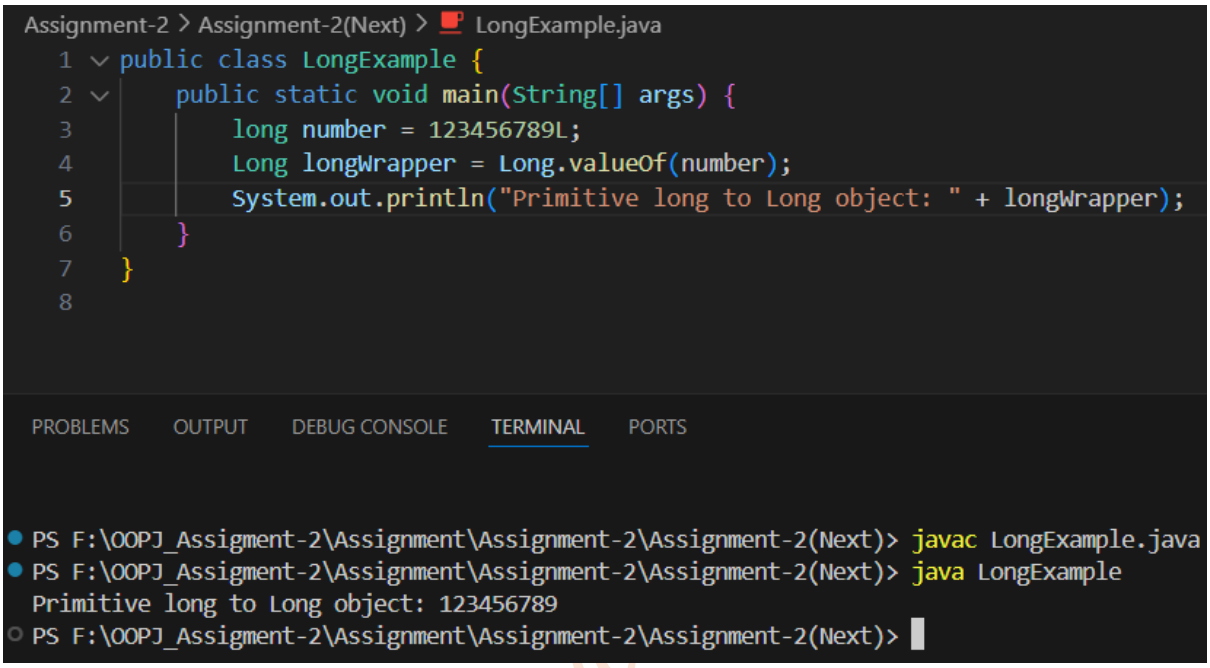
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `long` value using the `parseLong` method. (Hint: Use `Long.parseLong(String)`).



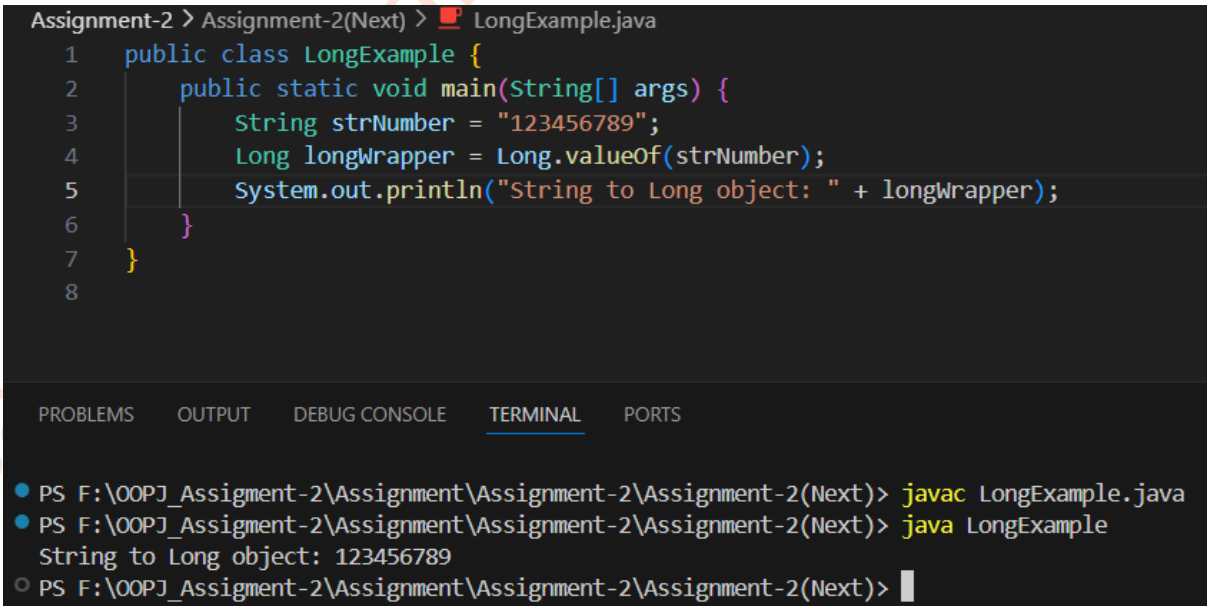
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `long` value. (Hint: `parseLong` method will throw a `NumberFormatException`).



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



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()`. (Hint: Use `Long.valueOf(String)`).



i. Declare two long variables with values `1123` and `9845`, and add them using a method from the `Long` class. (Hint: Use `Long.sum(long, long)`).

j. Declare two long variables with values `1122` and `5566`, and find the minimum and maximum values using the `Long` class. (Hint: Use `Long.min(long, long)` and `Long.max(long, long)`).

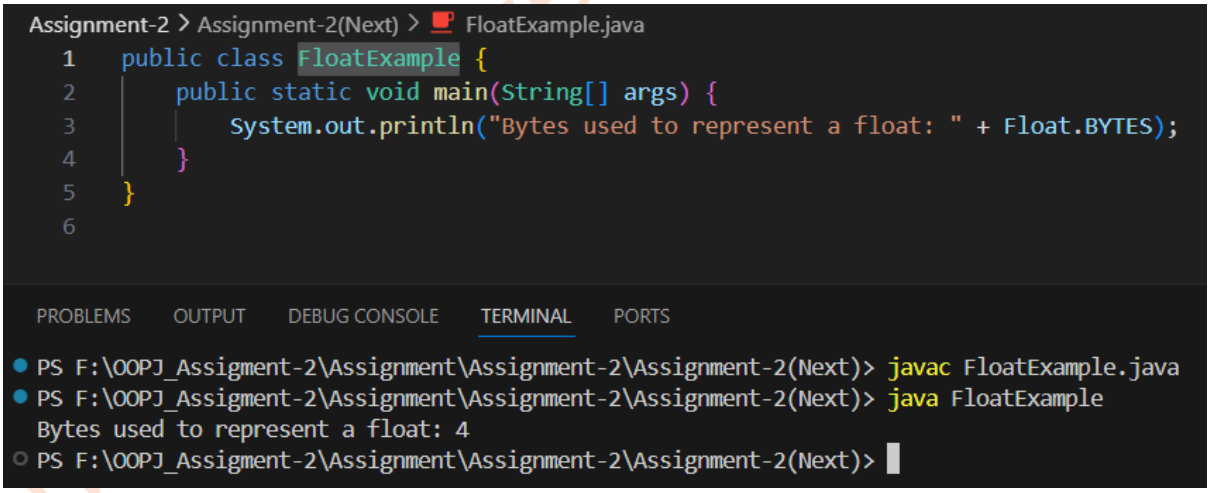
k. Declare a long variable with the value `7`. Convert it to binary, octal, and hexadecimal strings using methods from the `Long` class. (Hint: Use `Long.toBinaryString(long)`, `Long.toOctalString(long)`, and `Long.toHexString(long)`).

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

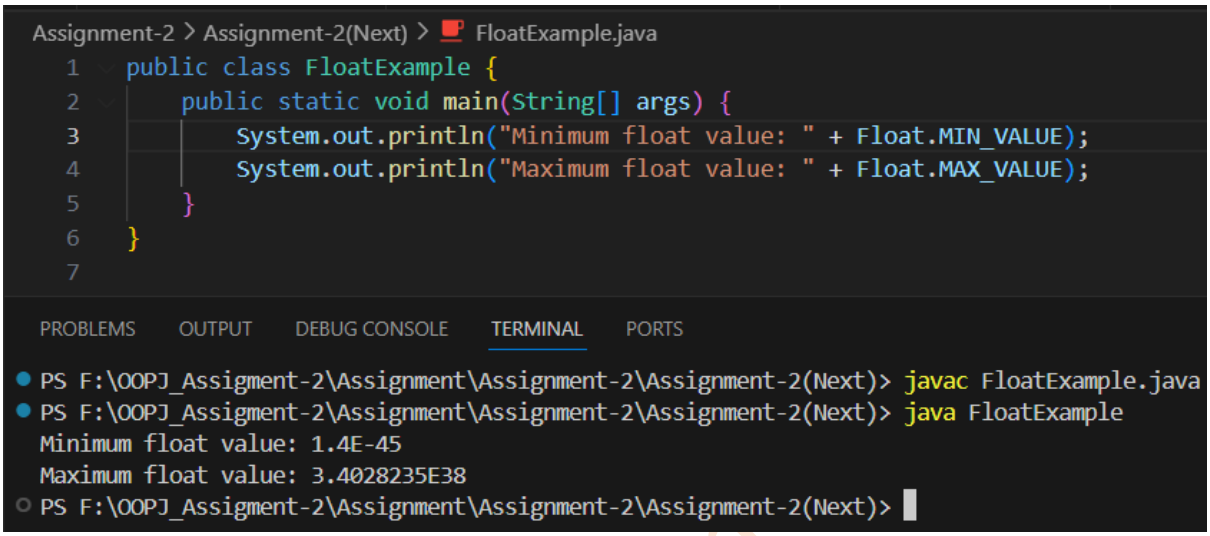
6. Working with `java.lang.Float`

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. (Hint: Use `Float.BYTES`).

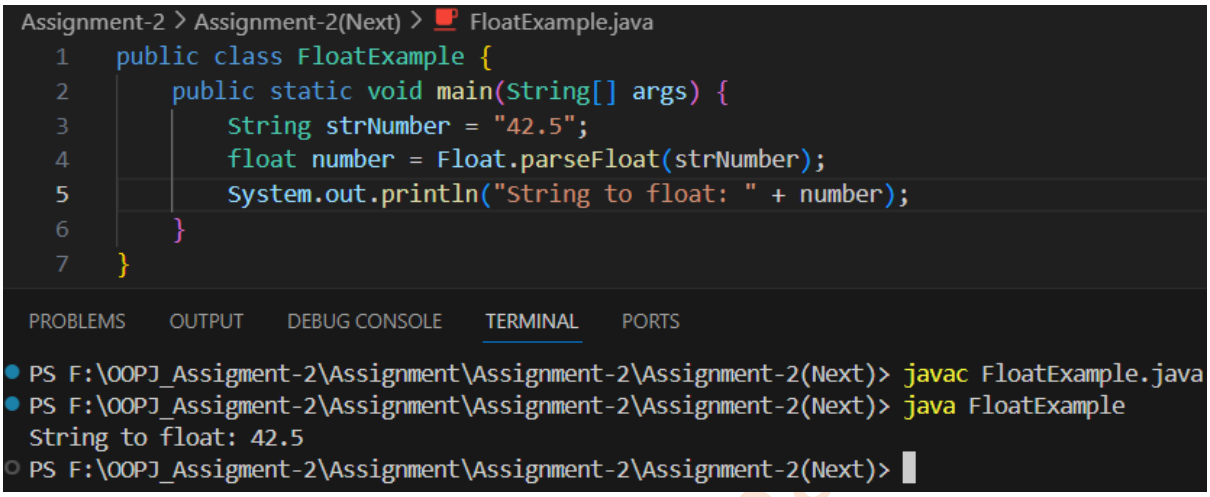


c. Write a program to find the minimum and maximum values of `float` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Float.MIN_VALUE` and `Float.MAX_VALUE`).

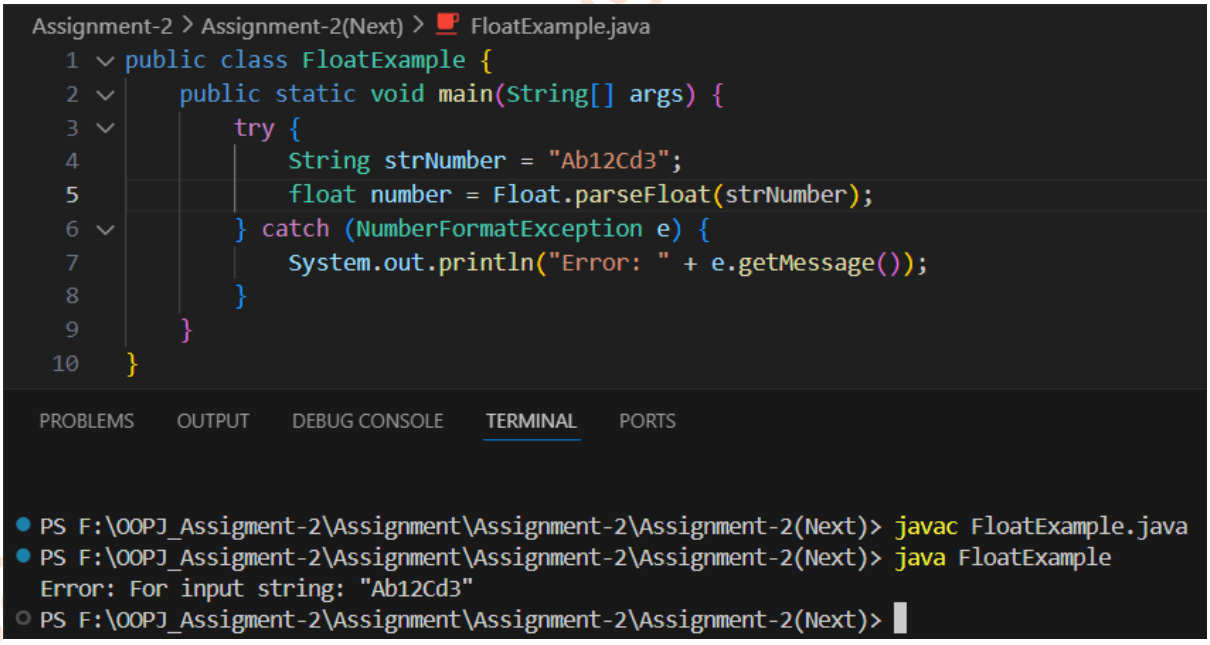


d. Declare a method-local variable `number` of type `float` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Float.toString(float)`).

e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `float` value using the `parseFloat` method. (Hint: Use `Float.parseFloat(String)`).



f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `float` value. (Hint: `parseFloat` method will throw a `NumberFormatException`).



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

Assignment-2 > Assignment-2(Next) > FloatExample.java

```java
public class FloatExample {
    public static void main(String[] args) {
        float number = 42.5f;
        Float floatWrapper = Float.valueOf(number);
        System.out.println("Primitive float to Float object: " + floatWrapper);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java FloatExample
Primitive float to Float object: 42.5
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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()`. (Hint: Use `Float.valueOf(String)`).

Assignment-2 > Assignment-2(Next) > FloatExample.java

```java
public class FloatExample {
    public static void main(String[] args) {
        String strNumber = "42.5";
        Float floatWrapper = Float.valueOf(strNumber);
        System.out.println("String to Float object: " + floatWrapper);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java FloatExample
String to Float object: 42.5
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

ASSIGNMENT NO.2

i. Declare two float variables with values `112.3` and `984.5`, and add them using a method from the `Float` class. (Hint: Use `Float.sum(float, float)`).

Assignment-2 > Assignment-2(Next) > FloatExample.java

```java
public class FloatExample {
    public static void main(String[] args) {
        float num1 = 112.3f;
        float num2 = 984.5f;
        float sum = Float.sum(num1, num2);
        System.out.println("Sum of 112.3 and 984.5: " + sum);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java FloatExample
Sum of 112.3 and 984.5: 1096.8
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

j. Declare two float variables with values `112.2` and `556.6`, and find the minimum and maximum values using the `Float` class. (Hint: Use `Float.min(float, float)` and `Float.max(float, float)`).

Assignment-2 > Assignment-2(Next) > FloatExample.java

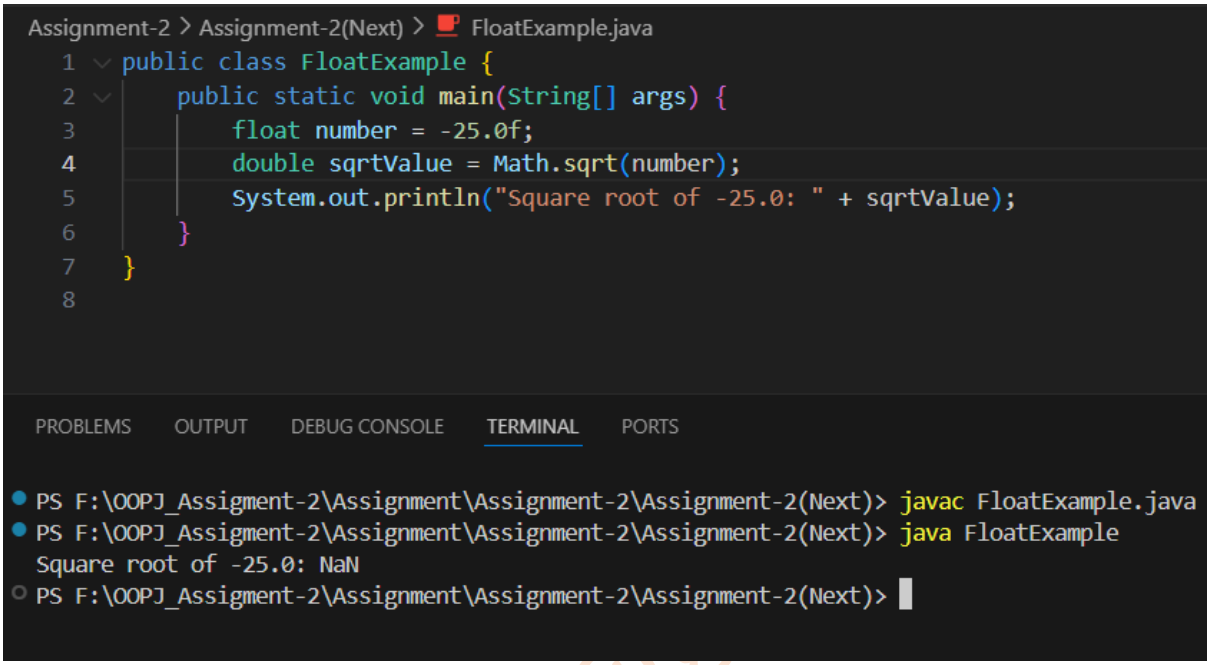
```java
public class FloatExample {
    public static void main(String[] args) {
        float num1 = 112.2f;
        float num2 = 556.6f;
        float min = Float.min(num1, num2);
        float max = Float.max(num1, num2);
        System.out.println("Min: " + min + ", Max: " + max);
    }
}

```

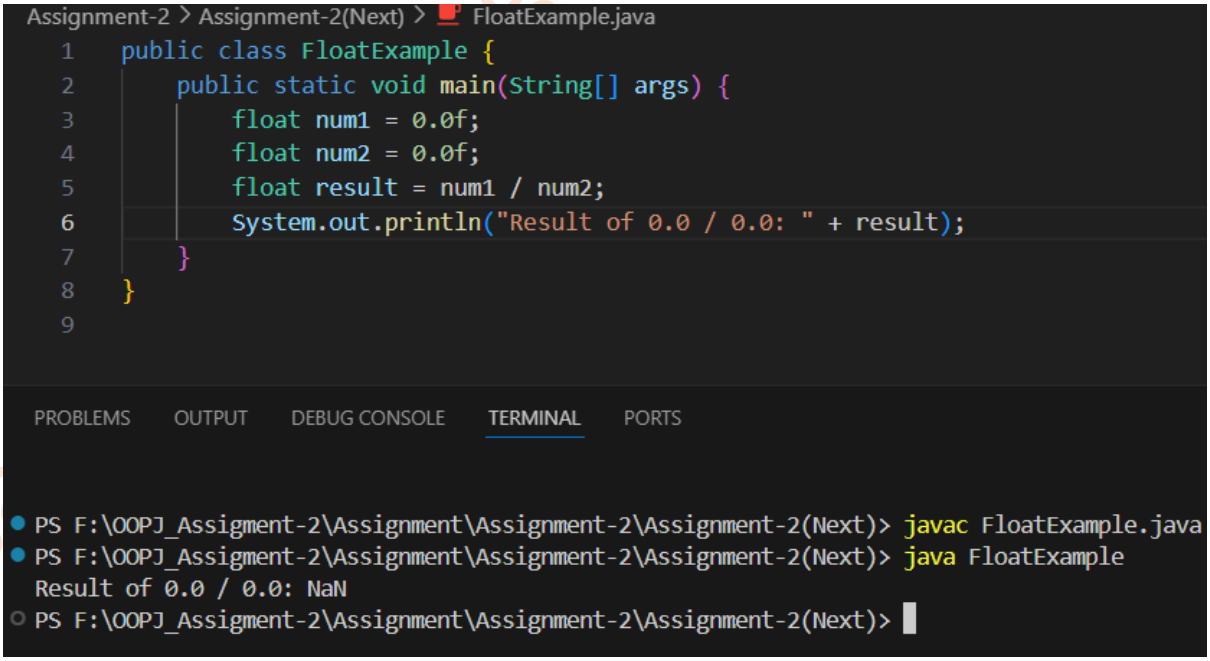
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java FloatExample
Min: 112.2, Max: 556.6
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

k. Declare a float variable with the value `-25.0f`. Find the square root of this value. (Hint: Use `Math.sqrt()` method).



l. Declare two float variables with the same value, `0.0f`, and divide them. (Hint: Observe the result and any special floating-point behavior).



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

Assignment-2 > Assignment-2(Next) > FloatExample.java

```java
public class FloatExample {
    public static void main(String[] args) {
        float number = 42.5f;

        int intValue = (int) number;
        System.out.println("Float to int: " + intValue);

        double doubleValue = (double) number;
        System.out.println("Float to double: " + doubleValue);

        double anotherDouble = 42.999;
        float doubleToFloat = (float) anotherDouble;
        System.out.println("Double to float: " + doubleToFloat);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac FloatExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java FloatExample
Float to int: 42
Float to double: 42.5
Double to float: 42.999
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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. (Hint: Use `Double.BYTES`).

```java
public class DoubleExample {
    public static void main(String[] args) {
        System.out.println("Bytes used to represent a double: " + Double.BYTES);
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Bytes used to represent a double: 8
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

c. Write a program to find the minimum and maximum values of `double` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Double.MIN_VALUE` and `Double.MAX_VALUE`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        System.out.println("Minimum double value: " + Double.MIN_VALUE);
        System.out.println("Maximum double value: " + Double.MAX_VALUE);
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Minimum double value: 4.9E-324
Maximum double value: 1.7976931348623157E308
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

d. Declare a method-local variable `number` of type `double` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Double.toString(double)`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double number = 42.5;
        String numberStr = Double.toString(number);
        System.out.println("Double to String: " + numberStr);
    }
}

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Double to String: 42.5
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `double` value using the `parseDouble` method. (Hint: Use `Double.parseDouble(String)`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

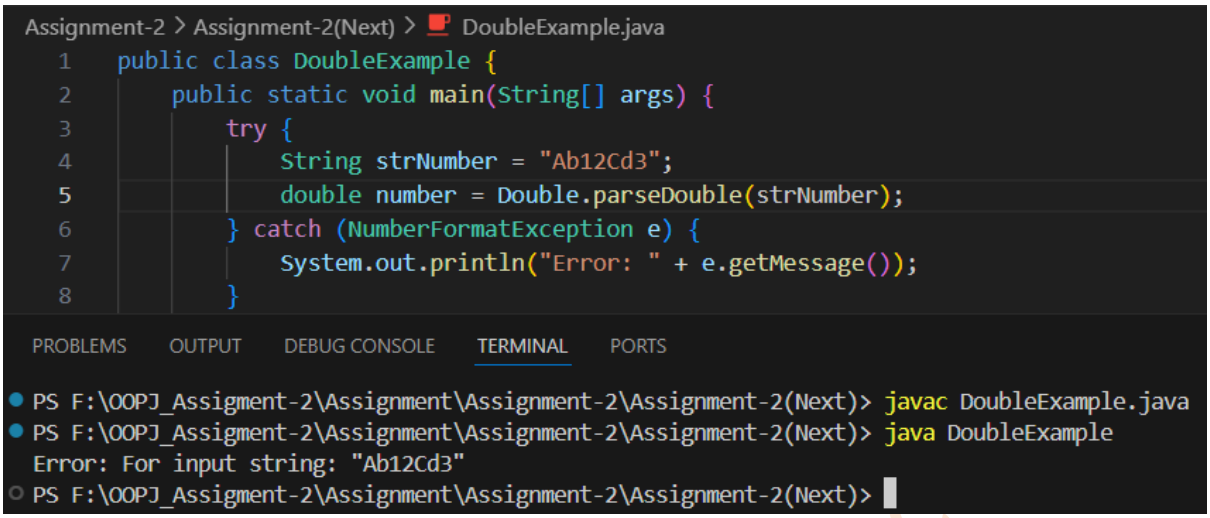
```java
public class DoubleExample {
    public static void main(String[] args) {
        String strNumber = "42.5";
        double number = Double.parseDouble(strNumber);
        System.out.println("String to double: " + number);
    }
}

```

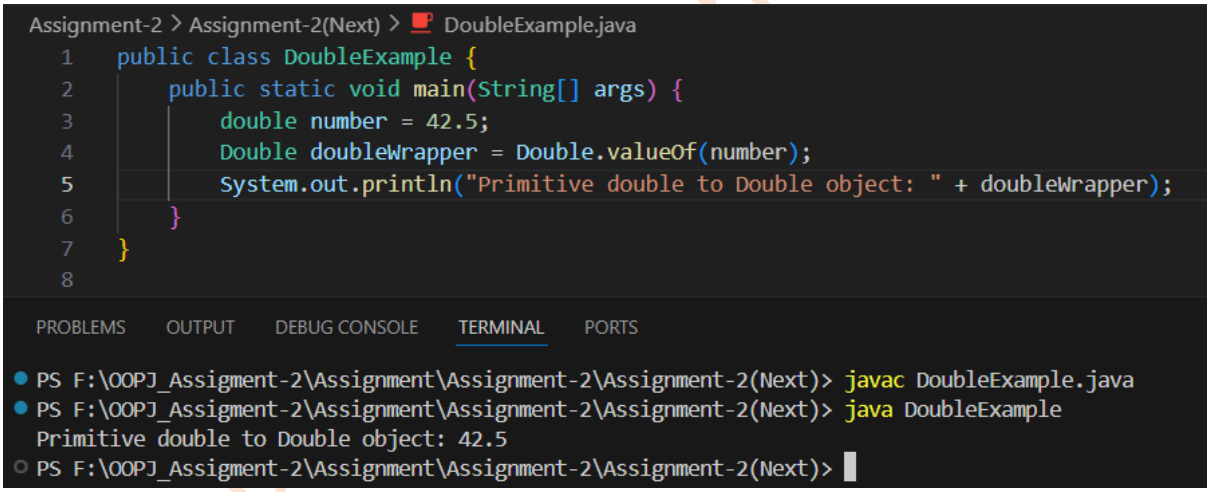
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
String to double: 42.5
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `double` value. (Hint: `parseDouble` method will throw a `NumberFormatException`).



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



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()`. (Hint: Use `Double.valueOf(String)`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        String strNumber = "42.5";
        Double doubleWrapper = Double.valueOf(strNumber);
        System.out.println("String to Double object: " + doubleWrapper);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
String to Double object: 42.5
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

i. Declare two double variables with values `112.3` and `984.5`, and add them using a method from the `Double` class. (Hint: Use `Double.sum(double, double)`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double num1 = 112.3;
        double num2 = 984.5;
        double sum = Double.sum(num1, num2);
        System.out.println("Sum of 112.3 and 984.5: " + sum);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Sum of 112.3 and 984.5: 1096.8
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

j. Declare two double variables with values `112.2` and `556.6`, and find the minimum and maximum values using the `Double` class. (Hint: Use `Double.min(double, double)` and `Double.max(double, double)`).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double num1 = 112.2;
        double num2 = 556.6;
        double min = Double.min(num1, num2);
        double max = Double.max(num1, num2);
        System.out.println("Min: " + min + ", Max: " + max);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Min: 112.2, Max: 556.6
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

k. Declare a double variable with the value `-25.0`. Find the square root of this value. (Hint: Use `Math.sqrt()` method).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double number = -25.0;
        double sqrtValue = Math.sqrt(number);
        System.out.println("Square root of -25.0: " + sqrtValue);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Square root of -25.0: NaN
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

ASSIGNMENT NO.2

l. Declare two double variables with the same value, `0.0`, and divide them. (Hint: Observe the result and any special floating-point behavior).

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double num1 = 0.0;
        double num2 = 0.0;
        double result = num1 / num2;
        System.out.println("Result of 0.0 / 0.0: " + result);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Result of 0.0 / 0.0: NaN
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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

Assignment-2 > Assignment-2(Next) > DoubleExample.java

```java
public class DoubleExample {
    public static void main(String[] args) {
        double number = 42.5;

        int intValue = (int) number;
        System.out.println("Double to int: " + intValue);

        float floatValue = (float) number;
        System.out.println("Double to float: " + floatValue);

        // Convert int to double
        int anotherInt = 42;
        double intToDouble = (double) anotherInt;
        System.out.println("Int to double: " + intToDouble); // Output: 42.0
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DoubleExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DoubleExample
Double to int: 42
Double to float: 42.5
Int to double: 42.0
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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()`).

Assignment-2 > Assignment-2(Next) > PrimitiveToStringExample.java

```java
public class PrimitiveToStringExample {
    public static void main(String[] args) {
        int intVar = 42;
        double doubleVar = 42.42;
        float floatVar = 42.42f;
        long longVar = 42424242L;
        boolean boolVar = true;
        byte byteVar = 42;
        short shortVar = 4242;
        char charVar = 'A';

        System.out.println("Integer to String (toString): " + Integer.toString(intVar));
        System.out.println("Double to String (toString): " + Double.toString(doubleVar));
        System.out.println("Float to String (toString): " + Float.toString(floatVar));
        System.out.println("Long to String (toString): " + Long.toString(longVar));
        System.out.println("Boolean to String (toString): " + Boolean.toString(boolVar));
        System.out.println("Byte to String (toString): " + Byte.toString(byteVar));
        System.out.println("Short to String (toString): " + Short.toString(shortVar));
        System.out.println("Char to String (toString): " + Character.toString(charVar));

        System.out.println("Integer to String (valueOf): " + String.valueOf(intVar));
        System.out.println("Double to String (valueOf): " + String.valueOf(doubleVar));
        System.out.println("Float to String (valueOf): " + String.valueOf(floatVar));
        System.out.println("Long to String (valueOf): " + String.valueOf(longVar));
        System.out.println("Boolean to String (valueOf): " + String.valueOf(boolVar));
        System.out.println("Byte to String (valueOf): " + String.valueOf(byteVar));
        System.out.println("Short to String (valueOf): " + String.valueOf(shortVar));
        System.out.println("Char to String (valueOf): " + String.valueOf(charVar));
    }
}
```

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac PrimitiveToStringExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java PrimitiveToStringExample
Integer to String (toString): 42
Double to String (toString): 42.42
Float to String (toString): 42.42
Long to String (toString): 42424242
Boolean to String (toString): true
Byte to String (toString): 42
Short to String (toString): 4242
Char to String (toString): A
Integer to String (valueOf): 42
Double to String (valueOf): 42.42
Float to String (valueOf): 42.42
Long to String (valueOf): 42424242
Boolean to String (valueOf): true
Byte to String (valueOf): 42
Short to String (valueOf): 4242
Char to String (valueOf): A
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

9. Default Values of Primitive Types

Declare variables of each primitive type as fields of a class and check their default values. (Note: Default values depend on whether the variables are instance variables or static variables).

Assignment-2 > Assignment-2(Next) > DefaultValueExample.java

```java
public class DefaultValueExample {
    int intVar;
    double doubleVar;
    float floatVar;
    long longVar;
    boolean boolVar;
    byte byteVar;
    short shortVar;
    char charVar;

    public static void main(String[] args) {
        DefaultValueExample obj = new DefaultValueExample();

        System.out.println("Default int value: " + obj.intVar);      // 0
        System.out.println("Default double value: " + obj.doubleVar); // 0.0
        System.out.println("Default float value: " + obj.floatVar);   // 0.0
        System.out.println("Default long value: " + obj.longVar);     // 0
        System.out.println("Default boolean value: " + obj.boolVar);  // false
        System.out.println("Default byte value: " + obj.byteVar);     // 0
        System.out.println("Default short value: " + obj.shortVar);   // 0
        System.out.println("Default char value: " + obj.charVar);     // '\u0000' (null character)
    }
}

```

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac DefaultValueExample.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java DefaultValueExample
Default int value: 0
Default double value: 0.0
Default float value: 0.0
Default long value: 0
Default boolean value: false
Default byte value: 0
Default short value: 0
Default char value:
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```

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. (Hint: Use `switch-case` for operations).

Assignment-2 > Assignment-2(Next) > DefaultValueExample.java

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

        try {
            // Parse the first and third arguments as integers
            int num1 = Integer.parseInt(args[0]);
            int num2 = Integer.parseInt(args[2]);
            char operator = args[1].charAt(0);

            switch (operator) {
                case '+':
                    System.out.println("Result: " + (num1 + num2));
                    break;
                case '-':
                    System.out.println("Result: " + (num1 - num2));
                    break;
                case '*':
                    System.out.println("Result: " + (num1 * num2));
                    break;
                case '/':
                    if (num2 != 0) {
                        System.out.println("Result: " + (num1 / num2));
                    } else {
                        System.out.println("Error: Division by zero!");
                    }
                    break;
                default:
                    System.out.println("Invalid operator. Use +, -, *, or /.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Error: Invalid number format.");
        }
    }
```

ASSIGNMENT NO.2

```
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> javac ArithmeticOperations.java
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ArithmeticOperations
Usage: java ArithmeticOperations <num1> <operator> <num2>
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)> java ArithmeticOperations 10 + 20
>>
Result: 30
PS F:\OOPJ_Assigment-2\Assignment\Assignment-2\Assignment-2(Next)>
```