ASSIGNMENT NO.2

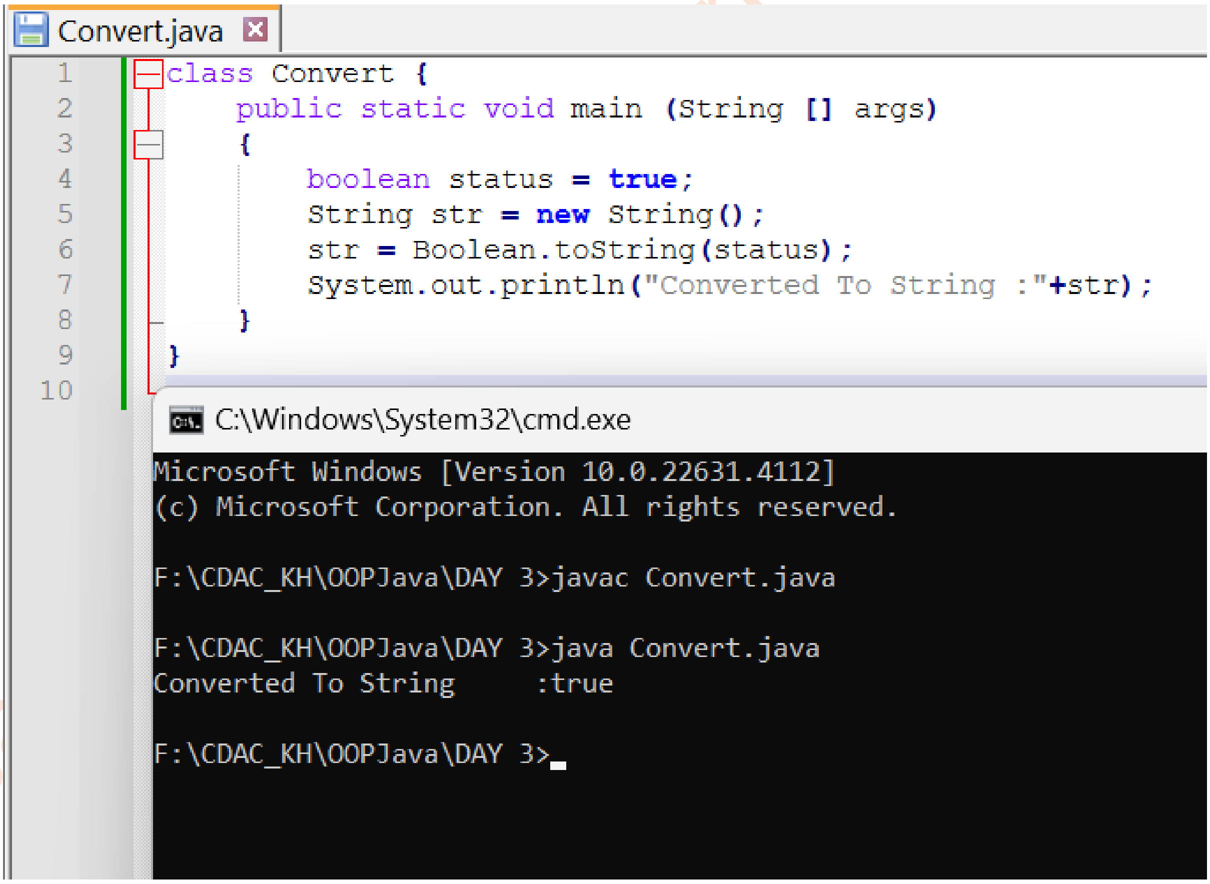
Note: Consider the following before starting the assignment:

- A **static field** declared inside a class is called a **class-level variable**. To access this variable, use the class name and the dot operator (e.g., `Integer.MAX_VALUE`).
- A **static method** defined inside a class is called a **class-level method**. To access this method, use the class name and the dot operator (e.g., `Integer.parseInt()`).
- When accessing static members within the same class, you do not need to use the class name.

1. Working with `java.lang.Boolean`

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

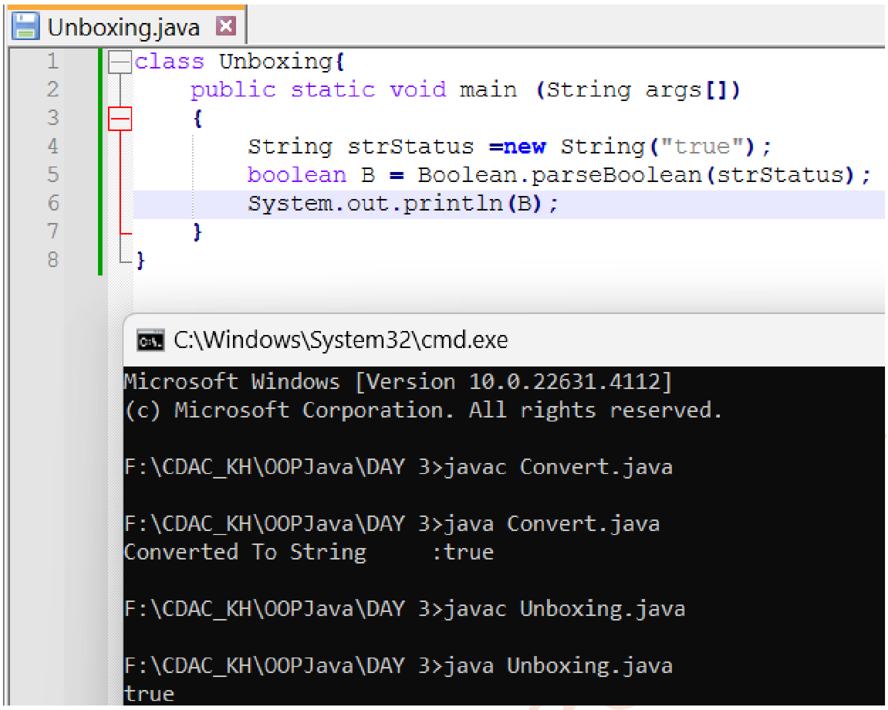
b. Declare a method-local variable `status` of type `boolean` with the value `true` and convert it to a `String` using the `toString` method. (Hint: Use `Boolean.toString(Boolean)`.



```java
class Convert {
    public static void main (String [] args)
    {
        boolean status = true;
        String str = new String();
        str = Boolean.toString(status);
        System.out.println("Converted To String :"+str);
    }
}
```

```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

F:\CDAC_KH\OOPJava\DAY 3>javac Convert.java

F:\CDAC_KH\OOPJava\DAY 3>java Convert.java
Converted To String     :true

F:\CDAC_KH\OOPJava\DAY 3>_
```

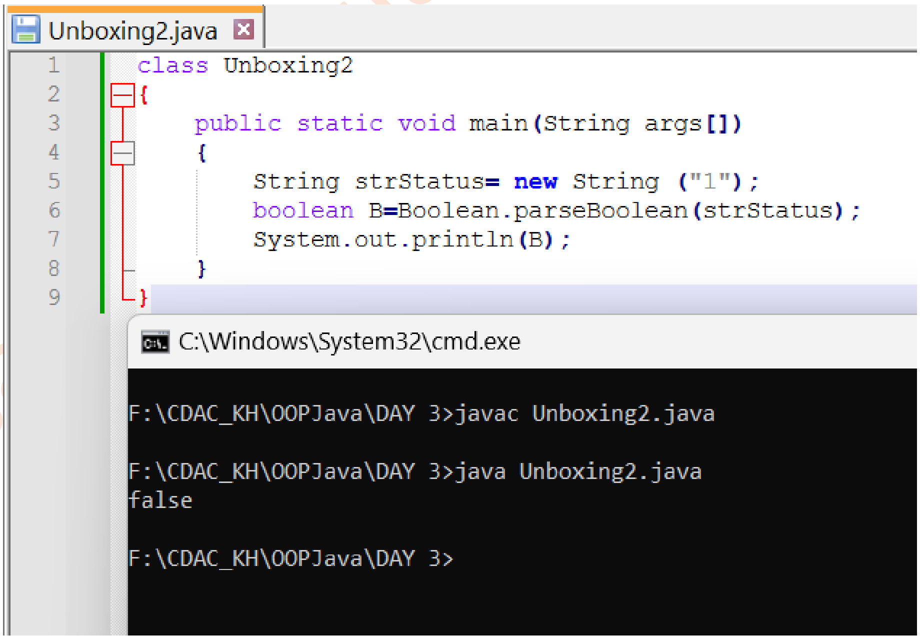
c. Declare a method-local variable `strStatus` of type `String` with the value `"true"` and convert it to a `boolean` using the `parseBoolean` method. (Hint: Use `Boolean.parseBoolean(String)`).



```java
class Unboxing{
        public static void main (String args[])
        {
            String strStatus =new String("true");
            boolean B = Boolean.parseBoolean(strStatus);
            System.out.println(B);
        }
}
```

```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

F:\CDAC_KH\OOPJava\DAY 3>javac Convert.java

F:\CDAC_KH\OOPJava\DAY 3>java Convert.java
Converted To String     :true

F:\CDAC_KH\OOPJava\DAY 3>javac Unboxing.java

F:\CDAC_KH\OOPJava\DAY 3>java Unboxing.java
true
```

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"`).

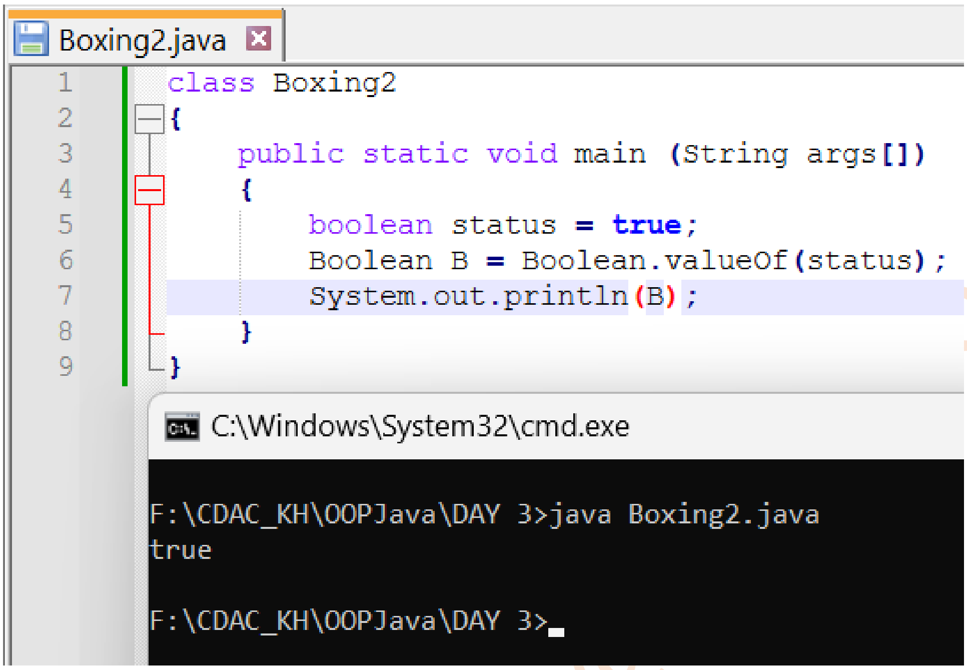


```java
class Unboxing2
{
    public static void main(String args[])
    {
        String strStatus= new String ("1");
        boolean B=Boolean.parseBoolean(strStatus);
        System.out.println(B);
    }
}
```

```
F:\CDAC_KH\OOPJava\DAY 3>javac Unboxing2.java

F:\CDAC_KH\OOPJava\DAY 3>java Unboxing2.java
false

F:\CDAC_KH\OOPJava\DAY 3>
```

Explanation: As Boolean stores only **true or false** in java but string strStatus passes the value “1” or “0” which result unexpected output.

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)`).



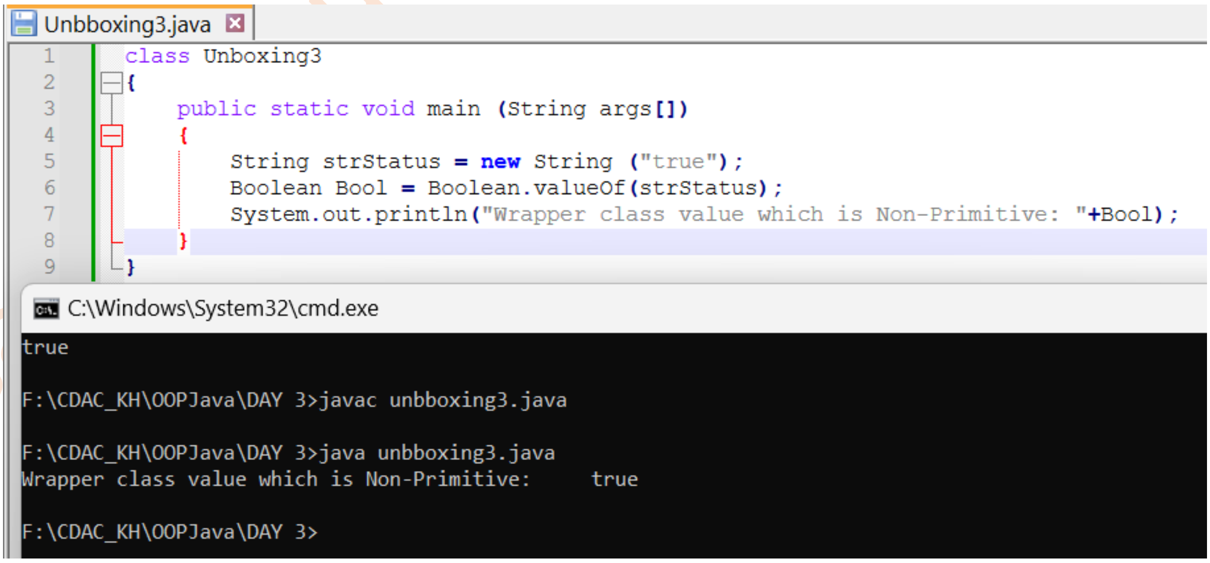
Boxing2.java

```java
class Boxing2
{
    public static void main (String args[])
    {
        boolean status = true;
        Boolean B = Boolean.valueOf(status);
        System.out.println(B);
    }
}
```

C:\Windows\System32\cmd.exe

```
F:\CDAC_KH\OOPJava\DAY 3>java Boxing2.java
true

F:\CDAC_KH\OOPJava\DAY 3>_
```

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)`).



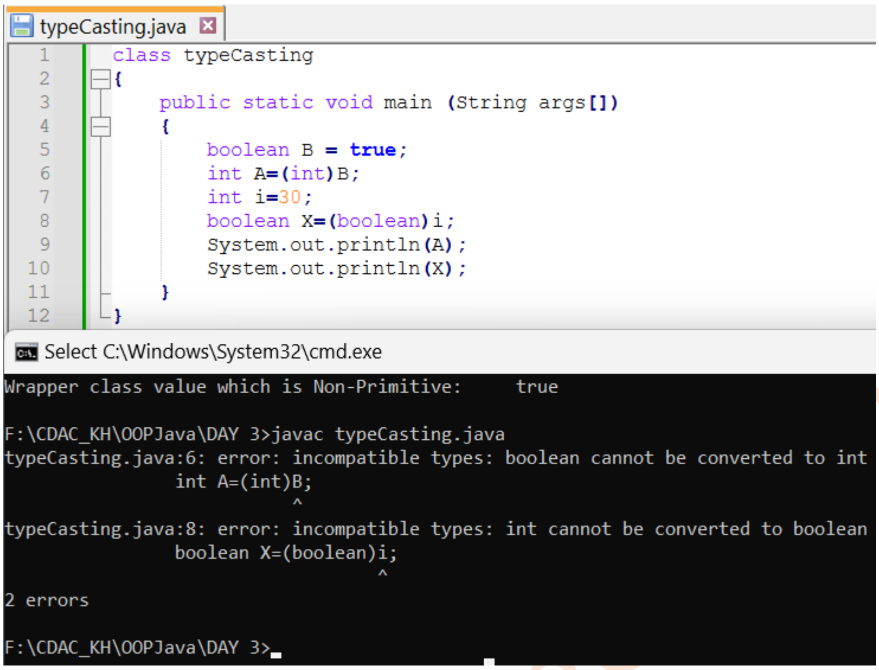
Unbboxing3.java

```java
class Unboxing3
{
    public static void main (String args[])
    {
        String strStatus = new String ("true");
        Boolean Bool = Boolean.valueOf(strStatus);
        System.out.println("Wrapper class value which is Non-Primitive:  "+Bool);
    }
}
```

C:\Windows\System32\cmd.exe

```
true

F:\CDAC_KH\OOPJava\DAY 3>javac unbboxing3.java

F:\CDAC_KH\OOPJava\DAY 3>java unbboxing3.java
Wrapper class value which is Non-Primitive:     true

F:\CDAC_KH\OOPJava\DAY 3>
```

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



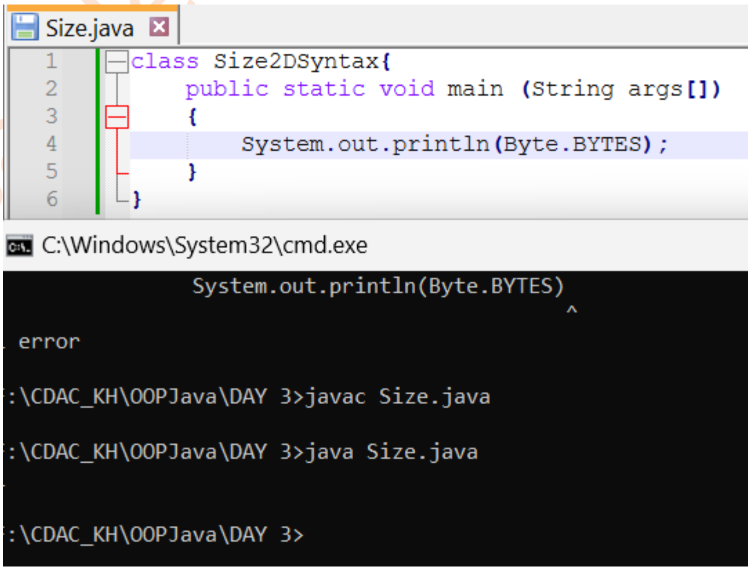
```
class typeCasting
{
    public static void main (String args[])
    {
        boolean B = true;
        int A=(int)B;
        int i=30;
        boolean X=(boolean)i;
        System.out.println(A);
        System.out.println(X);
    }
}
```

```
Wrapper class value which is Non-Primitive:     true

F:\CDAC_KH\OOPJava\DAY 3>javac typeCasting.java
typeCasting.java:6: error: incompatible types: boolean cannot be converted to int
                int A=(int)B;
                           ^
typeCasting.java:8: error: incompatible types: int cannot be converted to boolean
                boolean X=(boolean)i;
                                   ^
2 errors

F:\CDAC_KH\OOPJava\DAY 3>_
```

2. Working with `java.lang.Byte`

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

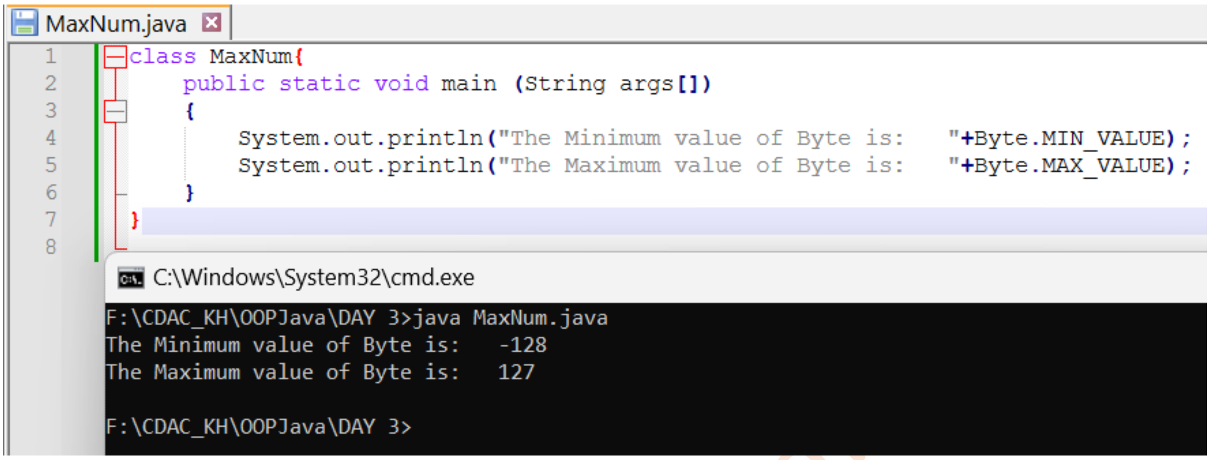
b. Write a program to test how many bytes are used to represent a `byte` value using the `BYTES` field. (Hint: Use `Byte.BYTES`).



```
class Size2DSyntax{
    public static void main (String args[])
    {
        System.out.println(Byte.BYTES);
    }
}
```

```
               System.out.println(Byte.BYTES)
                                             ^
 error

:\CDAC_KH\OOPJava\DAY 3>javac Size.java

:\CDAC_KH\OOPJava\DAY 3>java Size.java


:\CDAC_KH\OOPJava\DAY 3>
```

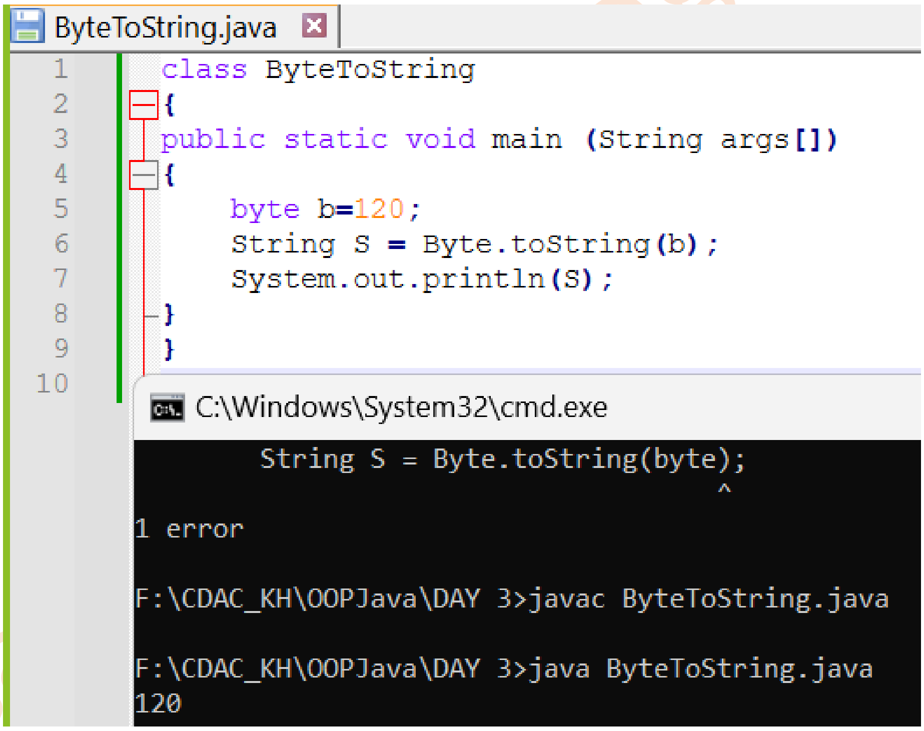
c. Write a program to find the minimum and maximum values of `byte` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Byte.MIN_VALUE` and `Byte.MAX_VALUE`).



```
class MaxNum{
    public static void main (String args[])
    {
        System.out.println("The Minimum value of Byte is:   "+Byte.MIN_VALUE);
        System.out.println("The Maximum value of Byte is:   "+Byte.MAX_VALUE);
    }
}
```

```
F:\CDAC_KH\OOPJava\DAY 3>java MaxNum.java
The Minimum value of Byte is:   -128
The Maximum value of Byte is:   127

F:\CDAC_KH\OOPJava\DAY 3>
```

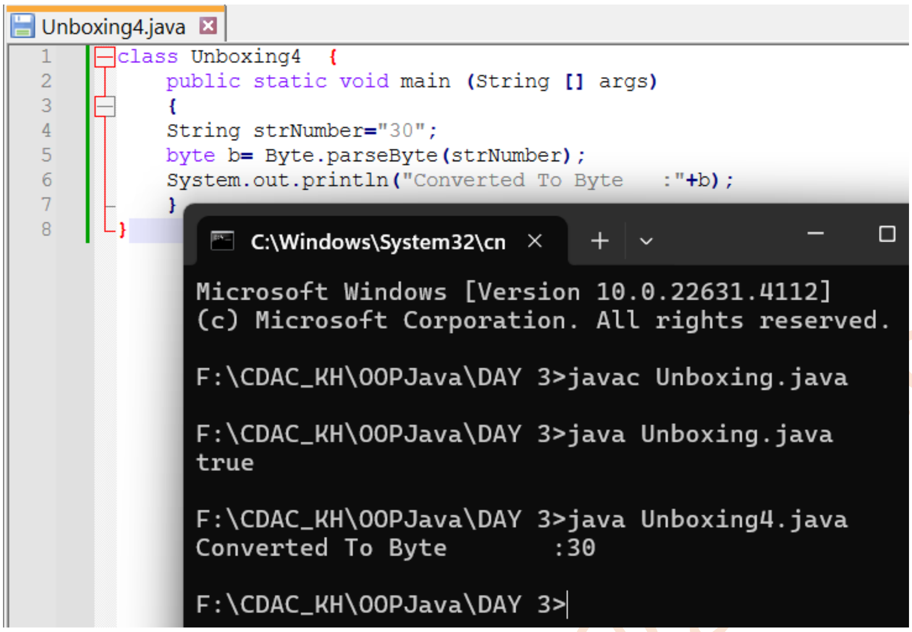
d. Declare a method-local variable `number` of type `byte` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Byte.toString(byte)`).



```
class ByteToString
{
public static void main (String args[])
{
    byte b=120;
    String S = Byte.toString(b);
    System.out.println(S);
}
}
```

```
        String S = Byte.toString(byte);
                                    ^

1 error

F:\CDAC_KH\OOPJava\DAY 3>javac ByteToString.java

F:\CDAC_KH\OOPJava\DAY 3>java ByteToString.java
120
```

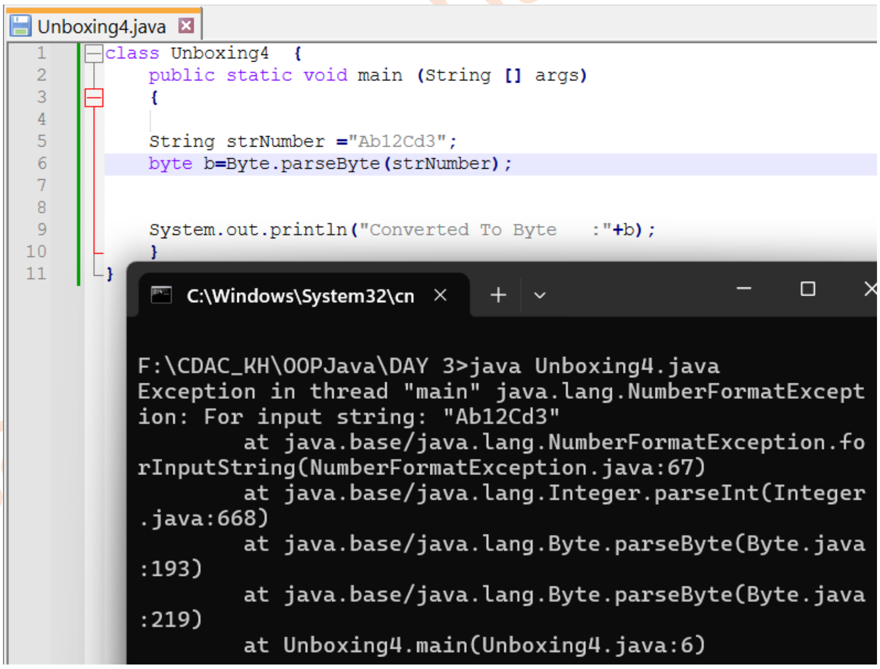
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `byte` value using the `parseByte` method. (Hint: Use `Byte.parseByte(String)`).



```java
class Unboxing4  {
	public static void main (String [] args)
	{
	String strNumber="30";
	byte b= Byte.parseByte(strNumber);
	System.out.println("Converted To Byte     :"+b);
	}
}
```

```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

F:\CDAC_KH\OOPJava\DAY 3>javac Unboxing.java

F:\CDAC_KH\OOPJava\DAY 3>java Unboxing.java
true

F:\CDAC_KH\OOPJava\DAY 3>java Unboxing4.java
Converted To Byte        :30

F:\CDAC_KH\OOPJava\DAY 3>
```

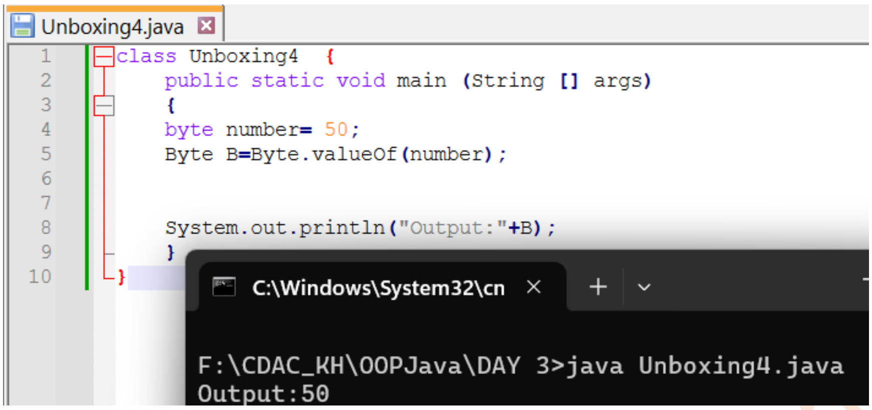
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `byte` value. (Hint: `parseByte` method will throw a `NumberFormatException`).



```java
class Unboxing4  {
	public static void main (String [] args)
	{

	String strNumber ="Ab12Cd3";
	byte b=Byte.parseByte(strNumber);


	System.out.println("Converted To Byte    :"+b);
	}
}
```

```
F:\CDAC_KH\OOPJava\DAY 3>java Unboxing4.java
Exception in thread "main" java.lang.NumberFormatExcept
ion: For input string: "Ab12Cd3"
        at java.base/java.lang.NumberFormatException.fo
rInputString(NumberFormatException.java:67)
        at java.base/java.lang.Integer.parseInt(Integer
.java:668)
        at java.base/java.lang.Byte.parseByte(Byte.java
:193)
        at java.base/java.lang.Byte.parseByte(Byte.java
:219)
        at Unboxing4.main(Unboxing4.java:6)
```

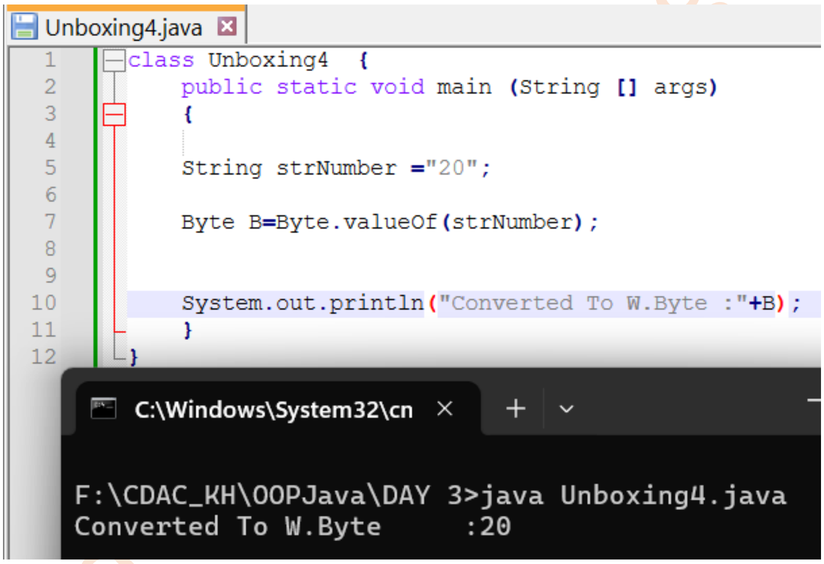
g. Declare a method-local variable `number` of type `byte` with some value and convert it to the corresponding wrapper class using `Byte.valueOf()`. (Hint: Use `Byte.valueOf(byte)`).



```java
class Unboxing4  {
    public static void main (String [] args)
    {
    byte number= 50;
    Byte B=Byte.valueOf(number);


    System.out.println("Output:"+B);
    }
}
```

```
F:\CDAC_KH\OOPJava\DAY 3>java Unboxing4.java
Output:50
```

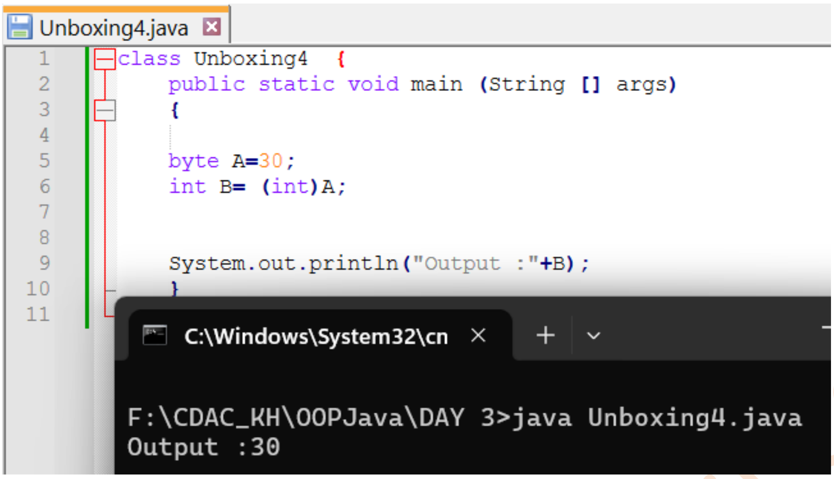
h. Declare a method-local variable `strNumber` of type `String` with some `byte` value and convert it to the corresponding wrapper class using `Byte.valueOf()`. (Hint: Use `Byte.valueOf(String)`).



```java
class Unboxing4  {
    public static void main (String [] args)
    {

    String strNumber ="20";

    Byte B=Byte.valueOf(strNumber);


    System.out.println("Converted To W.Byte :"+B);
    }
}
```

```
F:\CDAC_KH\OOPJava\DAY 3>java Unboxing4.java
Converted To W.Byte      :20
```

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



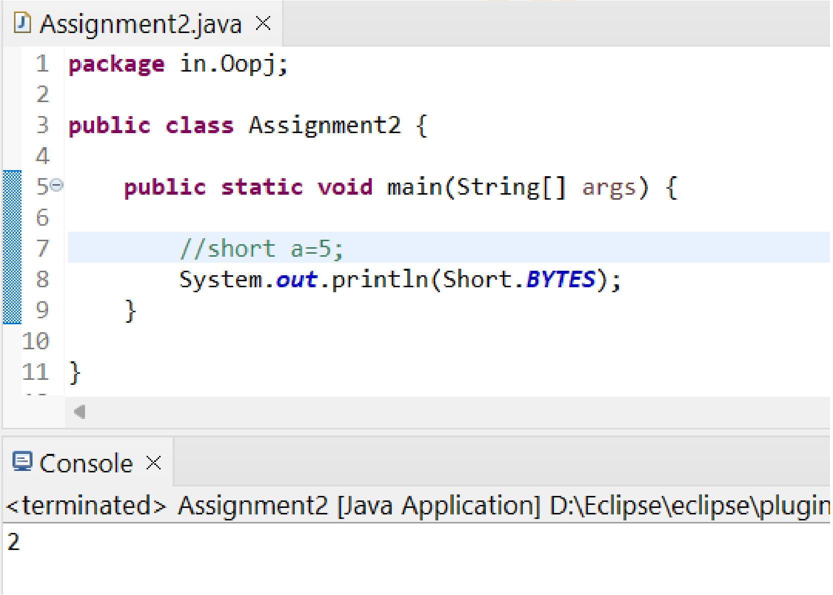
```
class Unboxing4  {
    public static void main (String [] args)
    {

    byte A=30;
    int B= (int)A;


    System.out.println("Output :"+B);
    }
```

```
F:\CDAC_KH\OOPJava\DAY 3>java Unboxing4.java
Output :30
```

3. Working with `java.lang.Short`

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

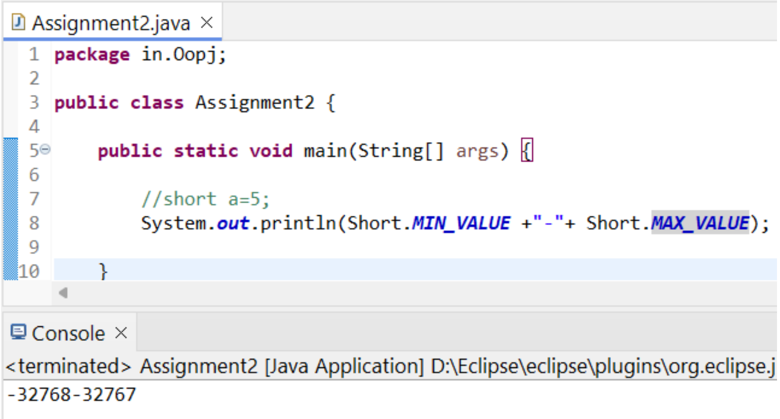
b. Write a program to test how many bytes are used to represent a `short` value using the `BYTES` field. (Hint: Use `Short.BYTES`).



```
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {

        //short a=5;
        System.out.println(Short.BYTES);
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugin
2
```

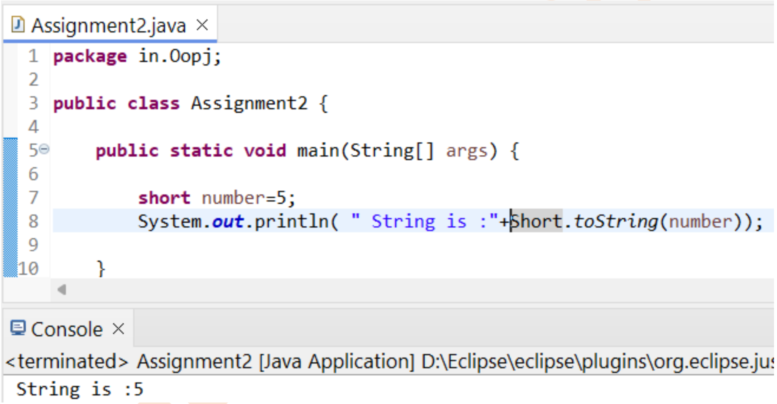
c. Write a program to find the minimum and maximum values of `short` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Short.MIN_VALUE` and `Short.MAX_VALUE`).



```java
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {

        //short a=5;
        System.out.println(Short.MIN_VALUE +"-"+ Short.MAX_VALUE);

    }
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.j
-32768-32767
```

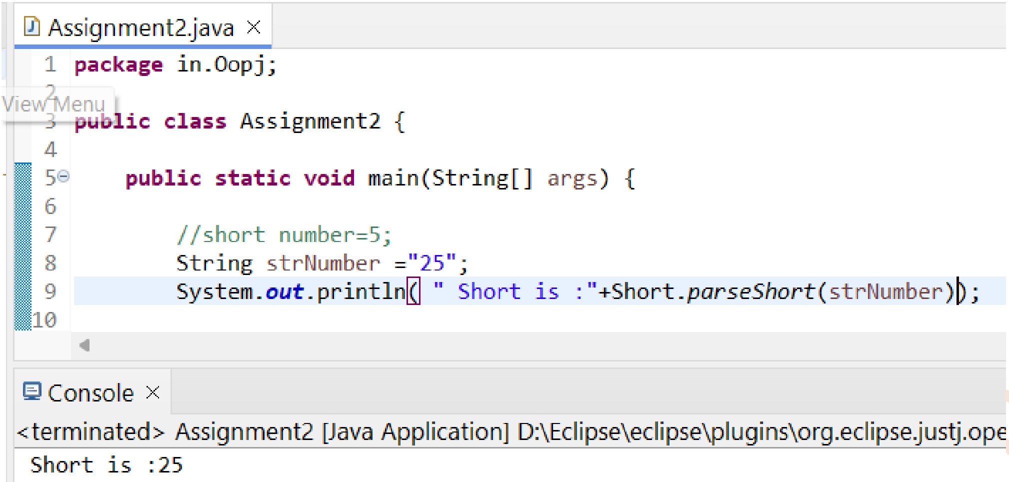
d. Declare a method-local variable `number` of type `short` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Short.toString(short)`).



```java
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {

        short number=5;
        System.out.println( " String is :"+Short.toString(number));

    }
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.ju
 String is :5
```

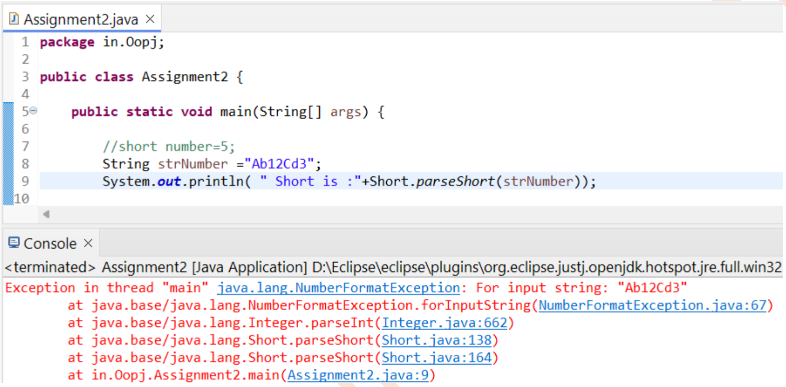
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `short` value using the `parseShort` method. (Hint: Use `Short.parseShort(String)`).



```
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {

        //short number=5;
        String strNumber ="25";
        System.out.println( " Short is :"+Short.parseShort(strNumber));

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.ope
Short is :25
```

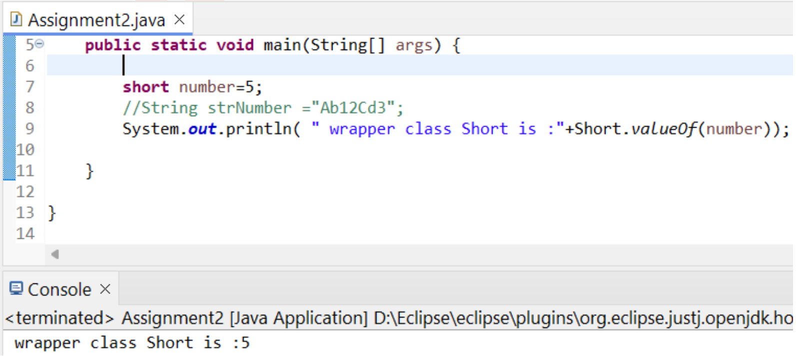
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `short` value. (Hint: `parseShort` method will throw a `NumberFormatException`).



```
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {

        //short number=5;
        String strNumber ="Ab12Cd3";
        System.out.println( " Short is :"+Short.parseShort(strNumber));

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
	at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
	at java.base/java.lang.Integer.parseInt(Integer.java:662)
	at java.base/java.lang.Short.parseShort(Short.java:138)
	at java.base/java.lang.Short.parseShort(Short.java:164)
	at in.Oopj.Assignment2.main(Assignment2.java:9)
```

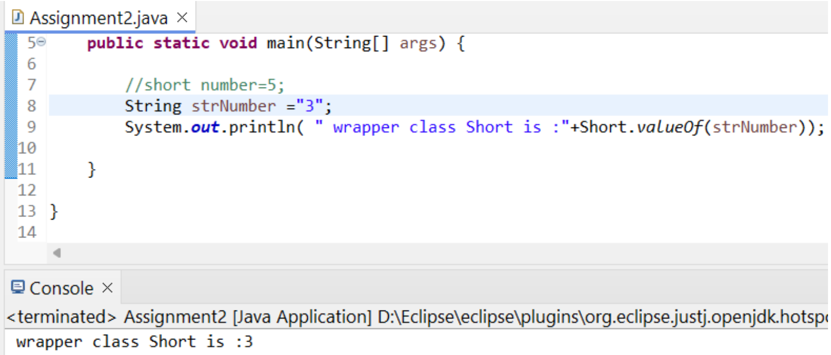
g. Declare a method-local variable `number` of type `short` with some value and convert it to the corresponding wrapper class using `Short.valueOf()`. (Hint: Use `Short.valueOf(short)`).



```
    public static void main(String[] args) {

        short number=5;
        //String strNumber ="Ab12Cd3";
        System.out.println( " wrapper class Short is :"+Short.valueOf(number));

    }

}

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.ho
wrapper class Short is :5
```

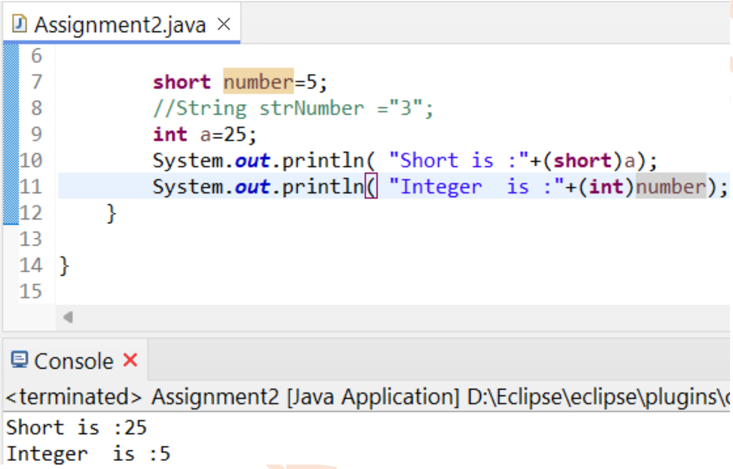
h. Declare a method-local variable `strNumber` of type `String` with some `short` value and convert it to the corresponding wrapper class using `Short.valueOf()`. (Hint: Use `Short.valueOf(String)`).



```
    public static void main(String[] args) {

        //short number=5;
        String strNumber ="3";
        System.out.println( " wrapper class Short is :"+Short.valueOf(strNumber));

    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotsp
 wrapper class Short is :3
```

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



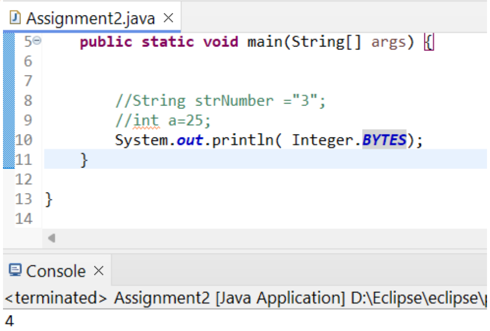
```
        short number=5;
        //String strNumber ="3";
        int a=25;
        System.out.println( "Short is :"+(short)a);
        System.out.println( "Integer  is :"+(int)number);
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\
Short is :25
Integer  is :5
```

4. Working with `java.lang.Integer`

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

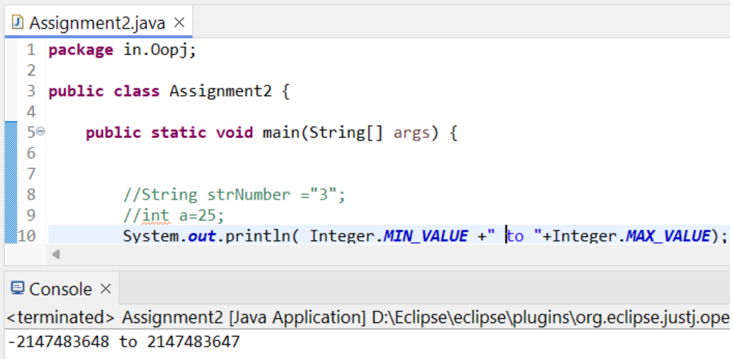
b. Write a program to test how many bytes are used to represent an `int` value using the `BYTES` field. (Hint: Use `Integer.BYTES`).



```
    public static void main(String[] args) {


        //String strNumber ="3";
        //int a=25;
        System.out.println( Integer.BYTES);
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\
4
```

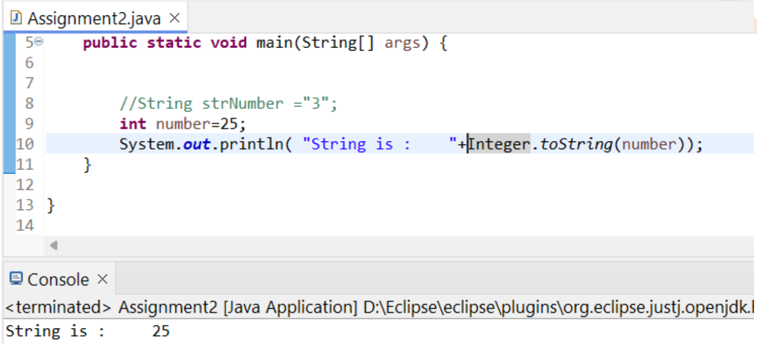
c. Write a program to find the minimum and maximum values of `int` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Integer.MIN_VALUE` and `Integer.MAX_VALUE`).



```java
package in.Oopj;

public class Assignment2 {

    public static void main(String[] args) {


        //String strNumber ="3";
        //int a=25;
        System.out.println( Integer.MIN_VALUE +" to "+Integer.MAX_VALUE);
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.ope
-2147483648 to 2147483647
```

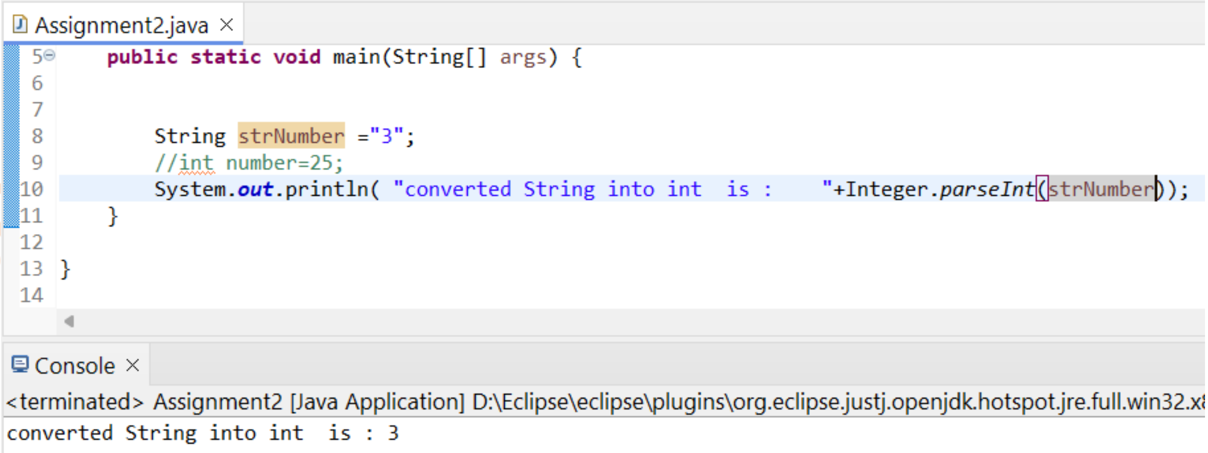
d. Declare a method-local variable `number` of type `int` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Integer.toString(int)`).



```java
    public static void main(String[] args) {


        //String strNumber ="3";
        int number=25;
        System.out.println( "String is :    "+Integer.toString(number));
    }

}

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.
String is :    25
```

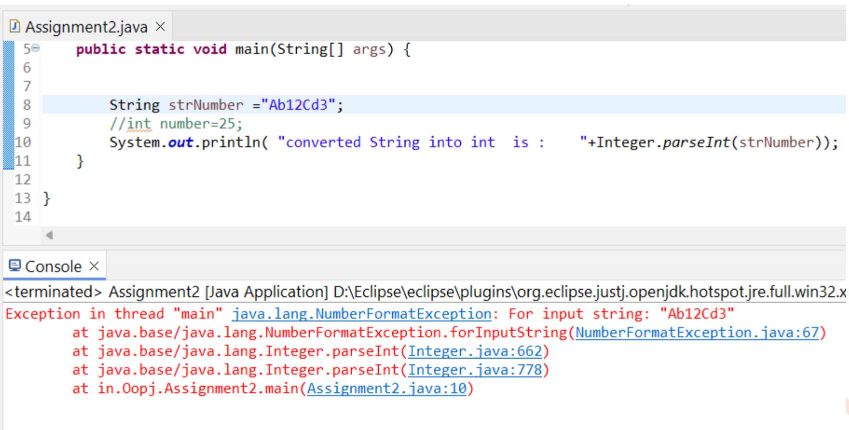
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to an `int` value using the `parseInt` method. (Hint: Use `Integer.parseInt(String)`).



```java
    public static void main(String[] args) {


        String strNumber ="3";
        //int number=25;
        System.out.println( "converted String into int  is :    "+Integer.parseInt(strNumber));
    }

}

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
converted String into int  is : 3
```

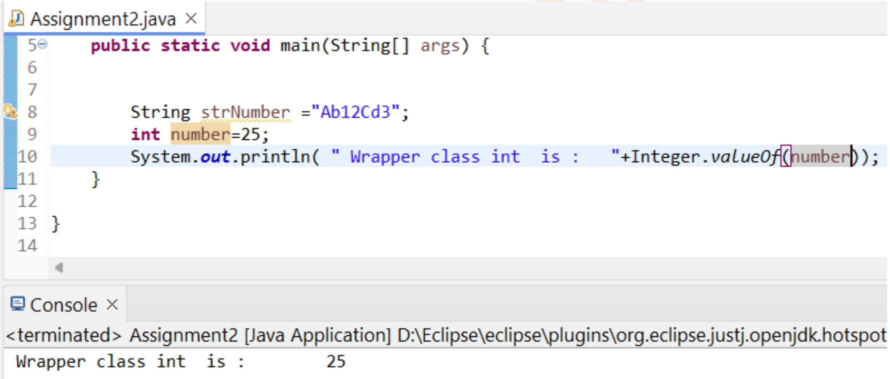
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to an `int` value. (Hint: `parseInt` method will throw a `NumberFormatException`).



```java
    public static void main(String[] args) {


        String strNumber ="Ab12Cd3";
        //int number=25;
        System.out.println( "converted String into int  is :    "+Integer.parseInt(strNumber));
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
	at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
	at java.base/java.lang.Integer.parseInt(Integer.java:662)
	at java.base/java.lang.Integer.parseInt(Integer.java:778)
	at in.Oopj.Assignment2.main(Assignment2.java:10)
```

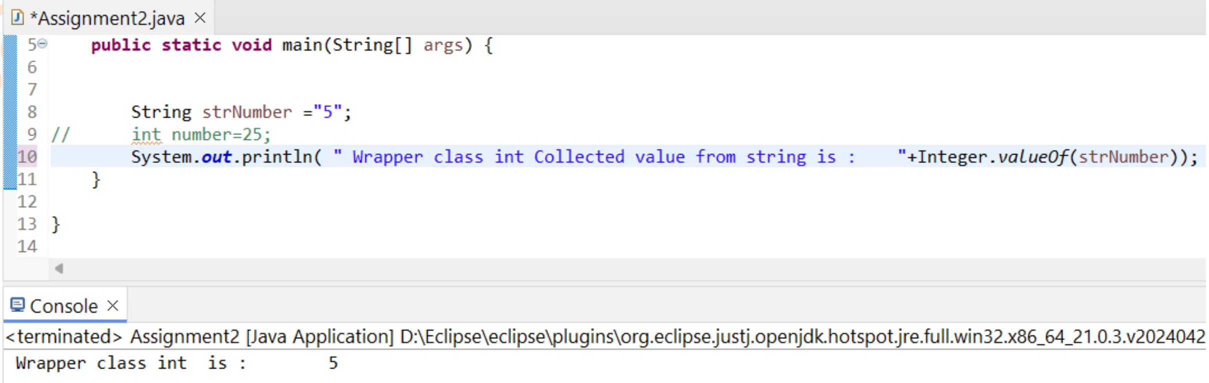
g. Declare a method-local variable `number` of type `int` with some value and convert it to the corresponding wrapper class using `Integer.valueOf()`. (Hint: Use `Integer.valueOf(int)`).



```java
    public static void main(String[] args) {


        String strNumber ="Ab12Cd3";
        int number=25;
        System.out.println( " Wrapper class int  is :   "+Integer.valueOf(number));
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot
 Wrapper class int  is :       25
```

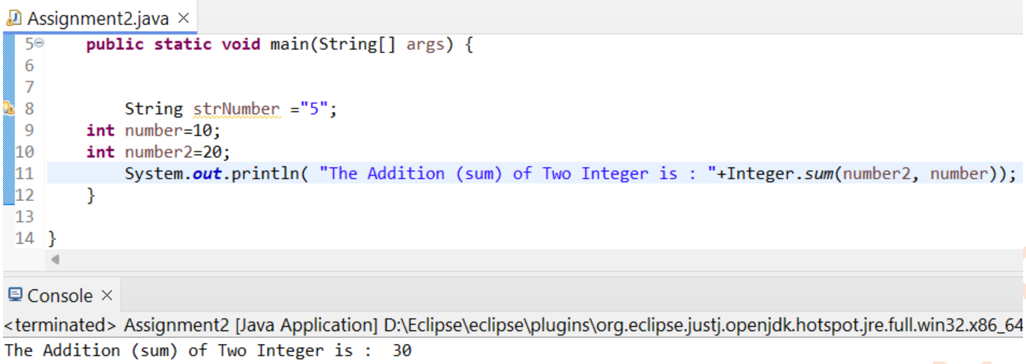
h. Declare a method-local variable `strNumber` of type `String` with some integer value and convert it to the corresponding wrapper class using `Integer.valueOf()`. (Hint: Use `Integer.valueOf(String)`).



```java
    public static void main(String[] args) {


        String strNumber ="5";
//      int number=25;
        System.out.println( " Wrapper class int Collected value from string is :    "+Integer.valueOf(strNumber));
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v2024042
 Wrapper class int  is :       5
```

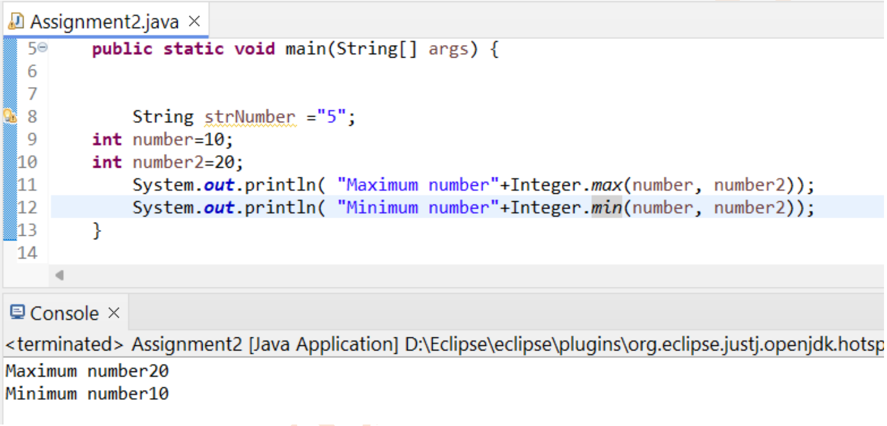
i. Declare two integer variables with values `10` and `20`, and add them using a method from the `Integer` class. (Hint: Use `Integer.sum(int, int)`).



```java
    public static void main(String[] args) {


        String strNumber ="5";
    int number=10;
    int number2=20;
        System.out.println( "The Addition (sum) of Two Integer is : "+Integer.sum(number2, number));
    }

}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
The Addition (sum) of Two Integer is :  30
```

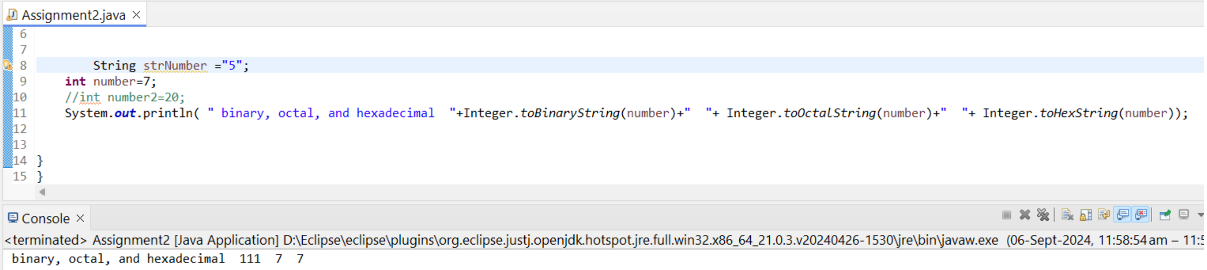
j. Declare two integer variables with values `10` and `20`, and find the minimum and maximum values using the `Integer` class. (Hint: Use `Integer.min(int, int)` and `Integer.max(int, int)`).



```java
    public static void main(String[] args) {


        String strNumber ="5";
    int number=10;
    int number2=20;
        System.out.println( "Maximum number"+Integer.max(number, number2));
        System.out.println( "Minimum number"+Integer.min(number, number2));
    }

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotsp
Maximum number20
Minimum number10
```

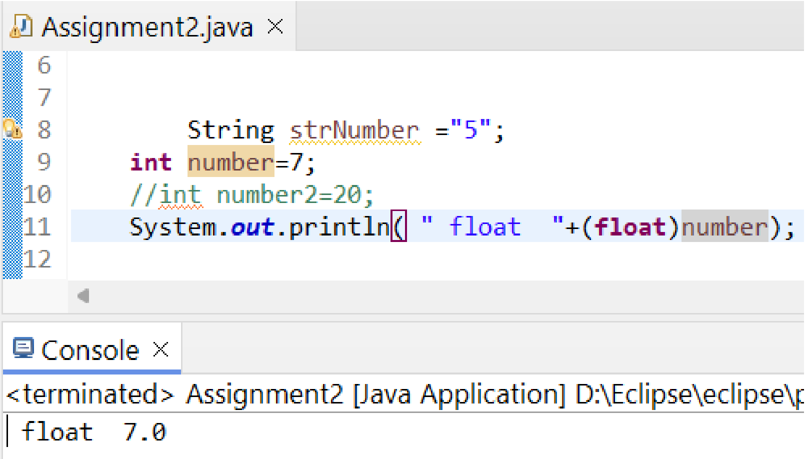
k. Declare an integer variable with the value `7`. Convert it to binary, octal, and hexadecimal strings using methods from the `Integer` class. (Hint: Use `Integer.toBinaryString(int)`, `Integer.toOctalString(int)`, and `Integer.toHexString(int)`).



```java


        String strNumber ="5";
    int number=7;
    //int number2=20;
    System.out.println( " binary, octal, and hexadecimal  "+Integer.toBinaryString(number)+"  "+ Integer.toOctalString(number)+"  "+ Integer.toHexString(number));



}
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe  (06-Sept-2024, 11:58:54 am – 11:5
 binary, octal, and hexadecimal  111  7  7
```

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



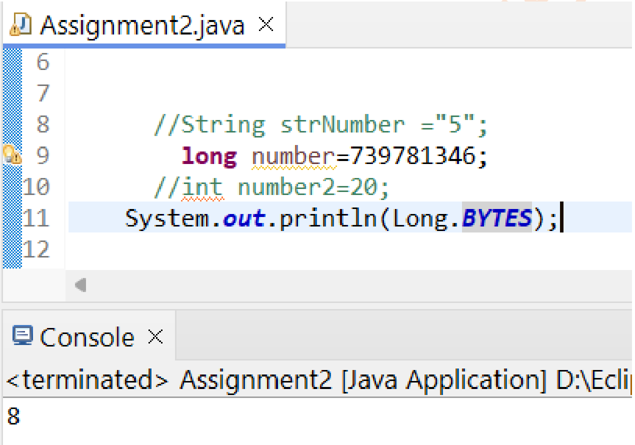
```java
        String strNumber ="5";
    int number=7;
    //int number2=20;
    System.out.println( " float   "+(float)number);
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\p
float  7.0
```

5. Working with `java.lang.Long`

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

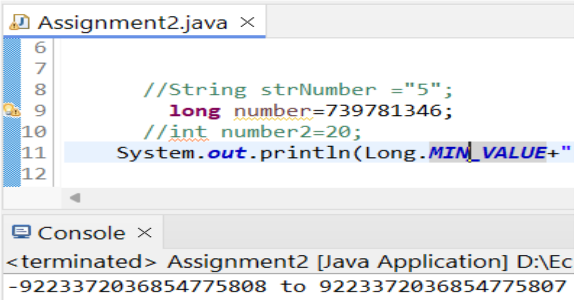
b. Write a program to test how many bytes are used to represent a `long` value using the `BYTES` field. (Hint: Use `Long.BYTES`).



```java
      //String strNumber ="5";
        long number=739781346;
      //int number2=20;
    System.out.println(Long.BYTES);
```

```
<terminated> Assignment2 [Java Application] D:\Ecli
8
```

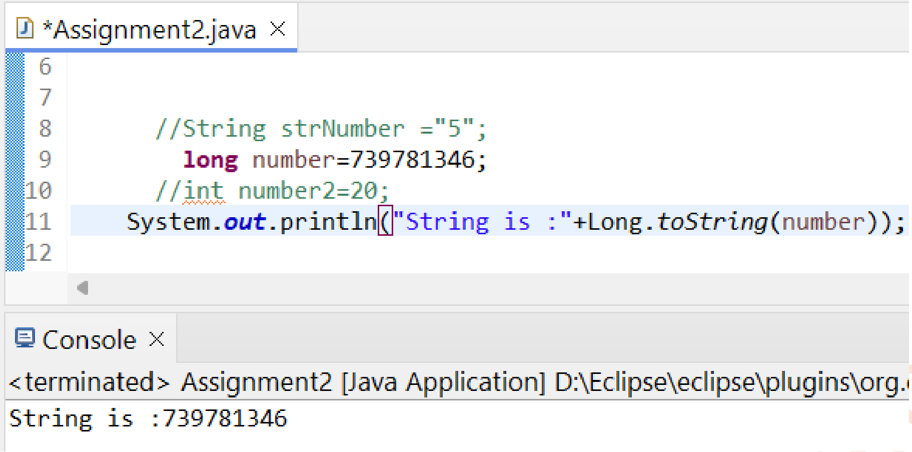
c. Write a program to find the minimum and maximum values of `long` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Long.MIN_VALUE` and `Long.MAX_VALUE`).



```java
      //String strNumber ="5";
        long number=739781346;
      //int number2=20;
    System.out.println(Long.MIN_VALUE+"
```

```
<terminated> Assignment2 [Java Application] D:\Ec
-9223372036854775808 to 9223372036854775807
```

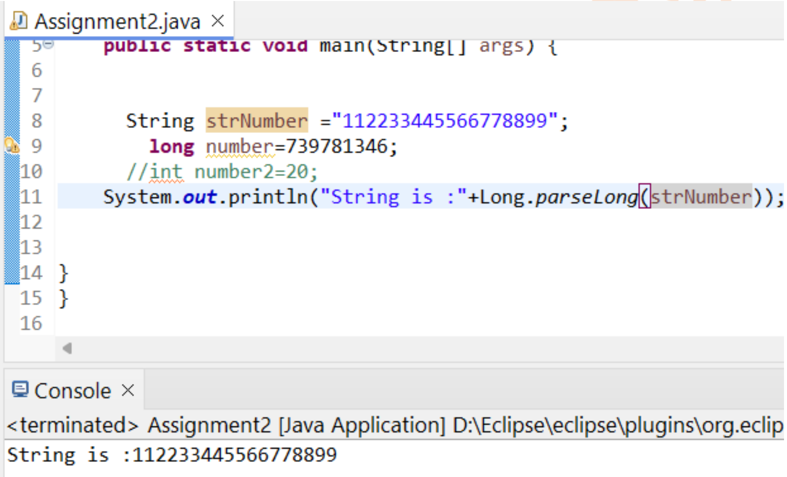
d. Declare a method-local variable `number` of type `long` with some value and convert it to a `String` using the `toString` method. (Hint: Use `Long.toString(long)`).



```
        //String strNumber ="5";
          long number=739781346;
        //int number2=20;
      System.out.println("String is :"+Long.toString(number));
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.
String is :739781346
```

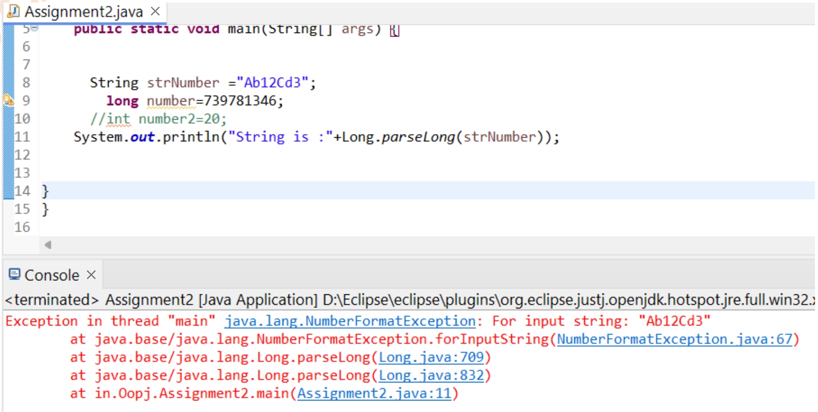
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `long` value using the `parseLong` method. (Hint: Use `Long.parseLong(String)`).



```
    public static void main(String[] args) {


        String strNumber ="112233445566778899";
          long number=739781346;
        //int number2=20;
    System.out.println("String is :"+Long.parseLong(strNumber));


}
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclip
String is :112233445566778899
```

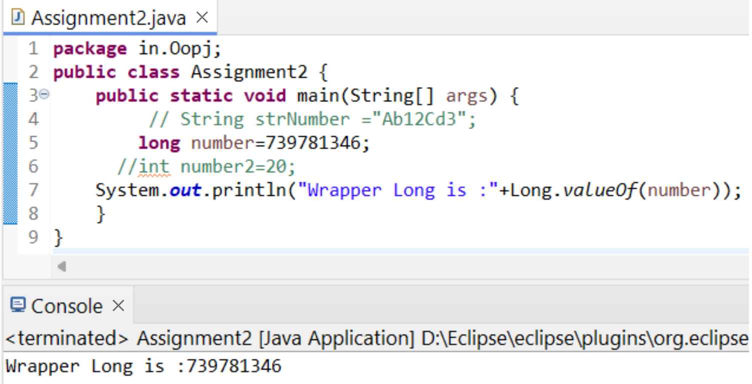
f. Declare a method-local variable `strNumber` of type `String` with the value `"Ab12Cd3"` and attempt to convert it to a `long` value. (Hint: `parseLong` method will throw a `NumberFormatException`).



```
    public static void main(String[] args) {


        String strNumber ="Ab12Cd3";
          long number=739781346;
        //int number2=20;
    System.out.println("String is :"+Long.parseLong(strNumber));


}
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
	at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
	at java.base/java.lang.Long.parseLong(Long.java:709)
	at java.base/java.lang.Long.parseLong(Long.java:832)
	at in.Oopj.Assignment2.main(Assignment2.java:11)
```

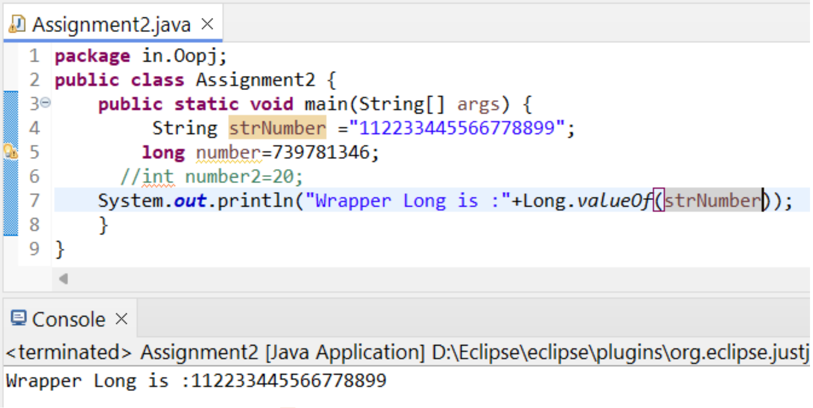
g. Declare a method-local variable `number` of type `long` with some value and convert it to the corresponding wrapper class using `Long.valueOf()`. (Hint: Use `Long.valueOf(long)`).



```
package in.Oopj;
public class Assignment2 {
    public static void main(String[] args) {
        // String strNumber ="Ab12Cd3";
        long number=739781346;
      //int number2=20;
    System.out.println("Wrapper Long is :"+Long.valueOf(number));
    }
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse
Wrapper Long is :739781346
```

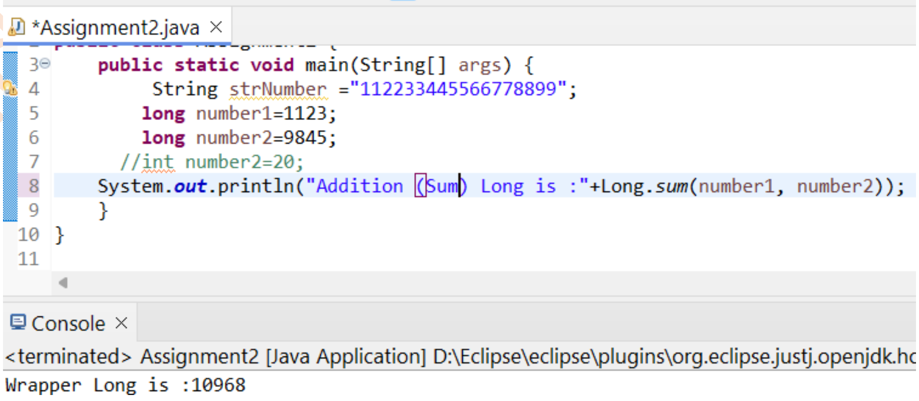
h. Declare a method-local variable `strNumber` of type `String` with some `long` value and convert it to the corresponding wrapper class using `Long.valueOf()`. (Hint: Use `Long.valueOf(String)`).



```
package in.Oopj;
public class Assignment2 {
    public static void main(String[] args) {
        String strNumber ="112233445566778899";
        long number=739781346;
      //int number2=20;
    System.out.println("Wrapper Long is :"+Long.valueOf(strNumber));
    }
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj
Wrapper Long is :112233445566778899
```

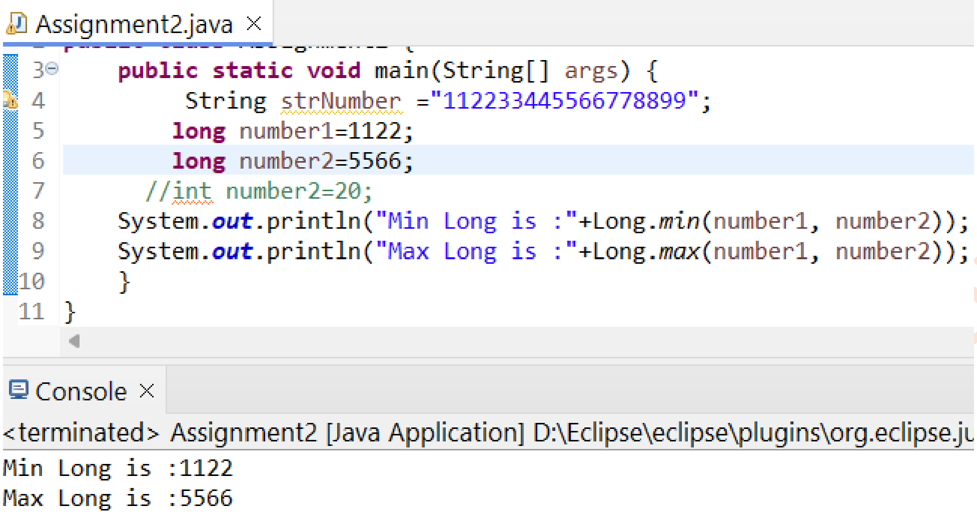
i. Declare two long variables with values `1123` and `9845`, and add them using a method from the `Long` class. (Hint: Use `Long.sum(long, long)`).



```
public class Assignment2 {
    public static void main(String[] args) {
        String strNumber ="112233445566778899";
        long number1=1123;
        long number2=9845;
      //int number2=20;
    System.out.println("Addition (Sum) Long is :"+Long.sum(number1, number2));
    }
}

```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hc
Wrapper Long is :10968
```

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)`).



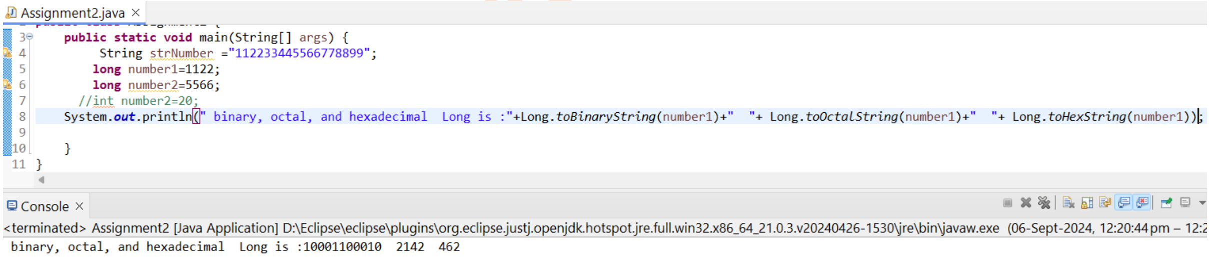
Assignment2.java

```
    public static void main(String[] args) {
          String strNumber ="112233445566778899";
        long number1=1122;
        long number2=5566;
      //int number2=20;
    System.out.println("Min Long is :"+Long.min(number1, number2));
    System.out.println("Max Long is :"+Long.max(number1, number2));
    }
}
```

Console

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.ju
Min Long is :1122
Max Long is :5566
```

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)`).



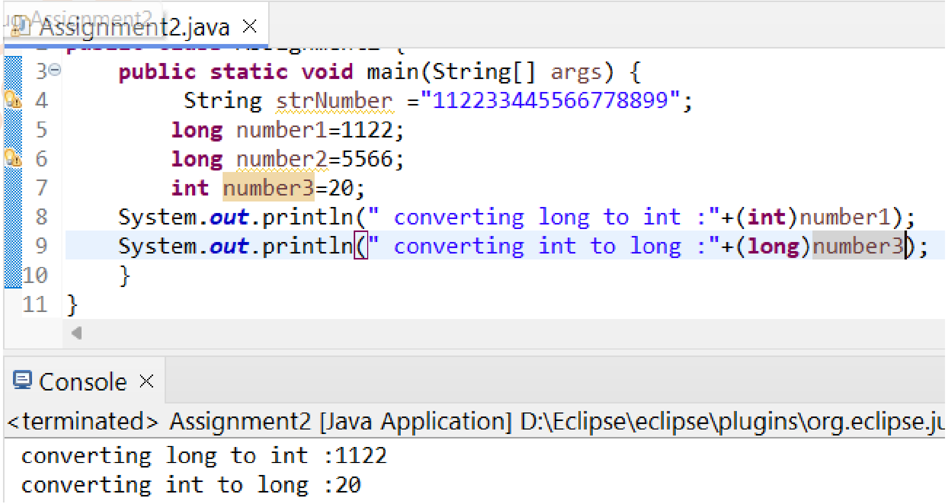
Assignment2.java

```
    public static void main(String[] args) {
          String strNumber ="112233445566778899";
        long number1=1122;
        long number2=5566;
      //int number2=20;
    System.out.println(" binary, octal, and hexadecimal  Long is :"+Long.toBinaryString(number1)+"  "+ Long.toOctalString(number1)+"  "+ Long.toHexString(number1));

    }
}
```

Console

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe  (06-Sept-2024, 12:20:44 pm – 12:2
 binary, octal, and hexadecimal  Long is :10001100010  2142  462
```

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



Assignment2.java

```
    public static void main(String[] args) {
          String strNumber ="112233445566778899";
        long number1=1122;
        long number2=5566;
        int number3=20;
    System.out.println(" converting long to int :"+(int)number1);
    System.out.println(" converting int to long :"+(long)number3);
    }
}
```

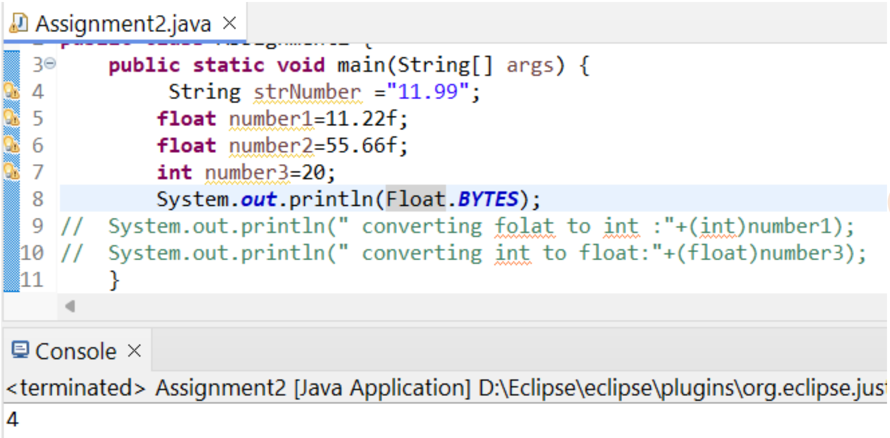
Console

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.ju
 converting long to int :1122
 converting int to long :20
```

6. Working with `java.lang.Float`

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

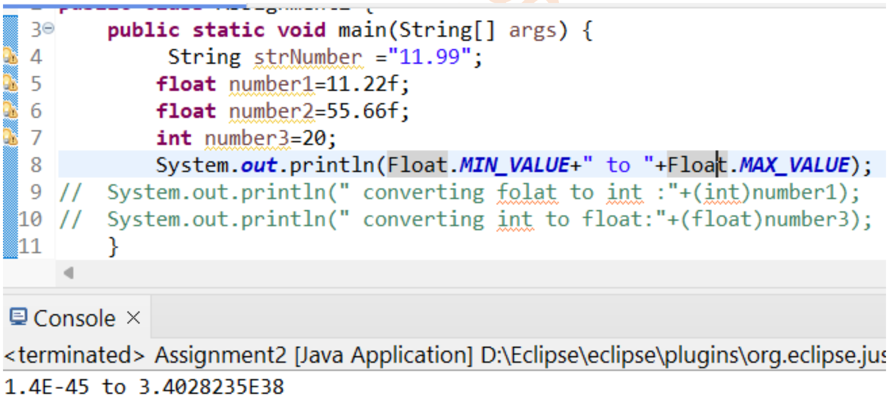
b. Write a program to test how many bytes are used to represent a `float` value using the `BYTES` field. (Hint: Use `Float.BYTES`).



```
    public static void main(String[] args) {
         String strNumber ="11.99";
        float number1=11.22f;
        float number2=55.66f;
        int number3=20;
        System.out.println(Float.BYTES);
//  System.out.println(" converting folat to int :"+(int)number1);
//  System.out.println(" converting int to float:"+(float)number3);
    }
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.jus
4
```

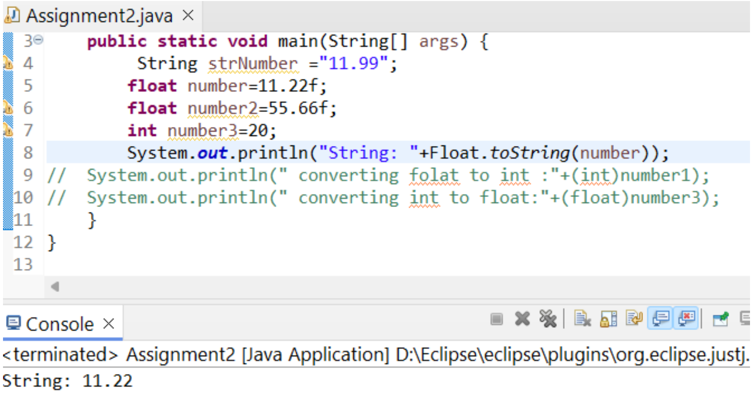
c. Write a program to find the minimum and maximum values of `float` using the `MIN_VALUE` and `MAX_VALUE` fields. (Hint: Use `Float.MIN_VALUE` and `Float.MAX_VALUE`).



```
    public static void main(String[] args) {
         String strNumber ="11.99";
        float number1=11.22f;
        float number2=55.66f;
        int number3=20;
        System.out.println(Float.MIN_VALUE+" to "+Float.MAX_VALUE);
//  System.out.println(" converting folat to int :"+(int)number1);
//  System.out.println(" converting int to float:"+(float)number3);
    }
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.jus
1.4E-45 to 3.4028235E38
```

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)`).



Assignment2.java ×

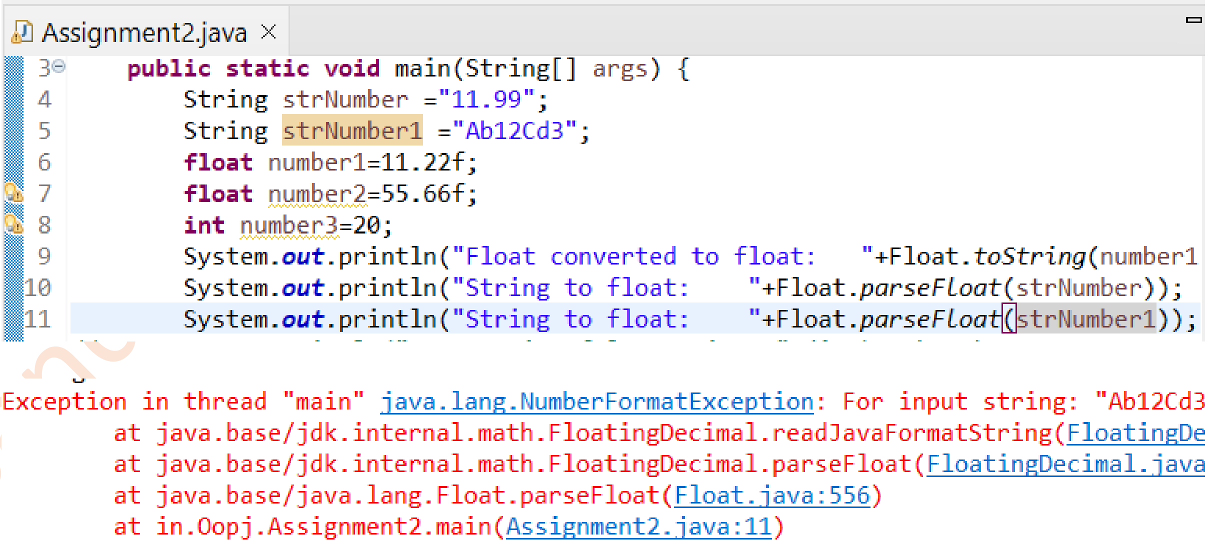
```
    public static void main(String[] args) {
         String strNumber ="11.99";
        float number=11.22f;
        float number2=55.66f;
        int number3=20;
        System.out.println("String:  "+Float.toString(number));
//  System.out.println(" converting folat to int :"+(int)number1);
//  System.out.println(" converting int to float:"+(float)number3);
    }
}
```

Console ×

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.
String: 11.22
```

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`).



Assignment2.java ×

```
    public static void main(String[] args) {
        String strNumber ="11.99";
        String strNumber1 ="Ab12Cd3";
        float number1=11.22f;
        float number2=55.66f;
        int number3=20;
        System.out.println("Float converted to float:   "+Float.toString(number1
        System.out.println("String to float:    "+Float.parseFloat(strNumber));
        System.out.println("String to float:    "+Float.parseFloat(strNumber1));
```

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3
	at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDe
	at java.base/jdk.internal.math.FloatingDecimal.parseFloat(FloatingDecimal.java
	at java.base/java.lang.Float.parseFloat(Float.java:556)
	at in.Oopj.Assignment2.main(Assignment2.java:11)
```

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)`).

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)`).

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)`).

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)`).

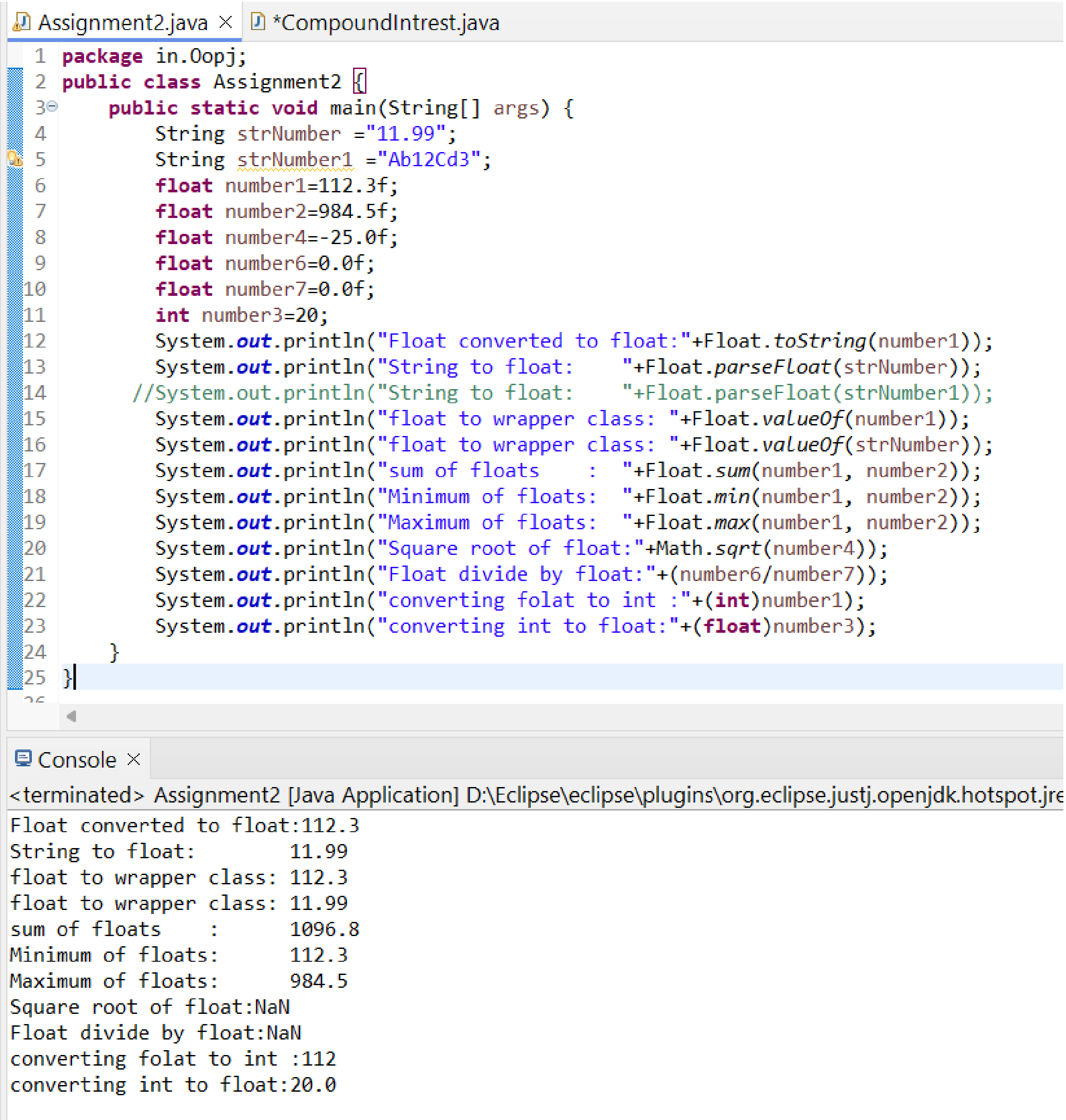
k. Declare a float variable with the value `-25.0f`. Find the square root of this value. (Hint: Use `Math.sqrt()` method).

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.

```
package in.Oopj;

public class Assignment2 {

        public static void main(String[] args) {

                String strNumber ="11.99";

                String strNumber1 ="Ab12Cd3";

                float number1=112.3f;

                float number2=984.5f;

                float number4=-25.0f;

                float number6=0.0f;

                float number7=0.0f;

           int number3=20;

           System.out.println("Float converted to float:"+Float.toString(number1));

           System.out.println("String to float:"+Float.parseFloat(strNumber));

          //System.out.println("String to float:          "+Float.parseFloat(strNumber1));

           System.out.println("float to wrapper class:
        "+Float.valueOf(number1));
```

```
                System.out.println("float to wrapper class: 
            "+Float.valueOf(strNumber));

                System.out.println("sum of floats    :         "+Float.sum(number1, 
number2));

                System.out.println("Minimum of floats:    "+Float.min(number1, 
number2));

                System.out.println("Maximum of floats:    "+Float.max(number1, 
number2));

                System.out.println("Square root of float:"+Math.sqrt(number4));

                System.out.println("Float divide by float:"+(number6/number7));

                System.out.println("converting folat to int :"+(int)number1);

        System.out.println("converting int to float:"+(float)number3);

            }

}
```



```java
package in.Oopj;
public class Assignment2 {
    public static void main(String[] args) {
        String strNumber ="11.99";
        String strNumber1 ="Ab12Cd3";
        float number1=112.3f;
        float number2=984.5f;
        float number4=-25.0f;
        float number6=0.0f;
        float number7=0.0f;
        int number3=20;
        System.out.println("Float converted to float:"+Float.toString(number1));
        System.out.println("String to float:    "+Float.parseFloat(strNumber));
      //System.out.println("String to float:    "+Float.parseFloat(strNumber1));
        System.out.println("float to wrapper class: "+Float.valueOf(number1));
        System.out.println("float to wrapper class: "+Float.valueOf(strNumber));
        System.out.println("sum of floats     :  "+Float.sum(number1, number2));
        System.out.println("Minimum of floats:  "+Float.min(number1, number2));
        System.out.println("Maximum of floats:  "+Float.max(number1, number2));
        System.out.println("Square root of float:"+Math.sqrt(number4));
        System.out.println("Float divide by float:"+(number6/number7));
        System.out.println("converting folat to int :"+(int)number1);
        System.out.println("converting int to float:"+(float)number3);
    }
}
```

```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre
Float converted to float:112.3
String to float:        11.99
float to wrapper class: 112.3
float to wrapper class: 11.99
sum of floats     :     1096.8
Minimum of floats:      112.3
Maximum of floats:      984.5
Square root of float:NaN
Float divide by float:NaN
converting folat to int :112
converting int to float:20.0
```

7. Working with `java.lang.Double`

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

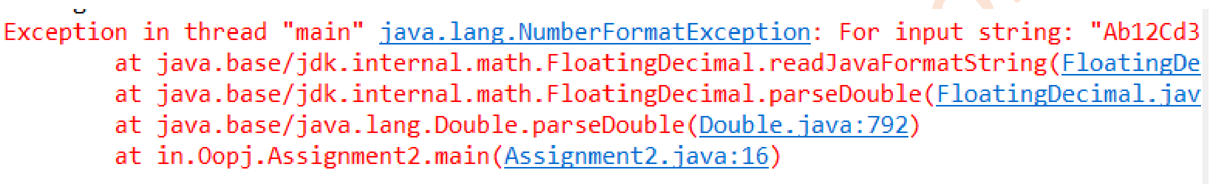
b. Write a program to test how many bytes are used to represent a `double` value using the `BYTES` field. (Hint: Use `Double.BYTES`).

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`).

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)`).

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)`).

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`).



```
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3
	at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDe
	at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.jav
	at java.base/java.lang.Double.parseDouble(Double.java:792)
	at in.Oopj.Assignment2.main(Assignment2.java:16)
```

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)`).

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)`).

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)`).

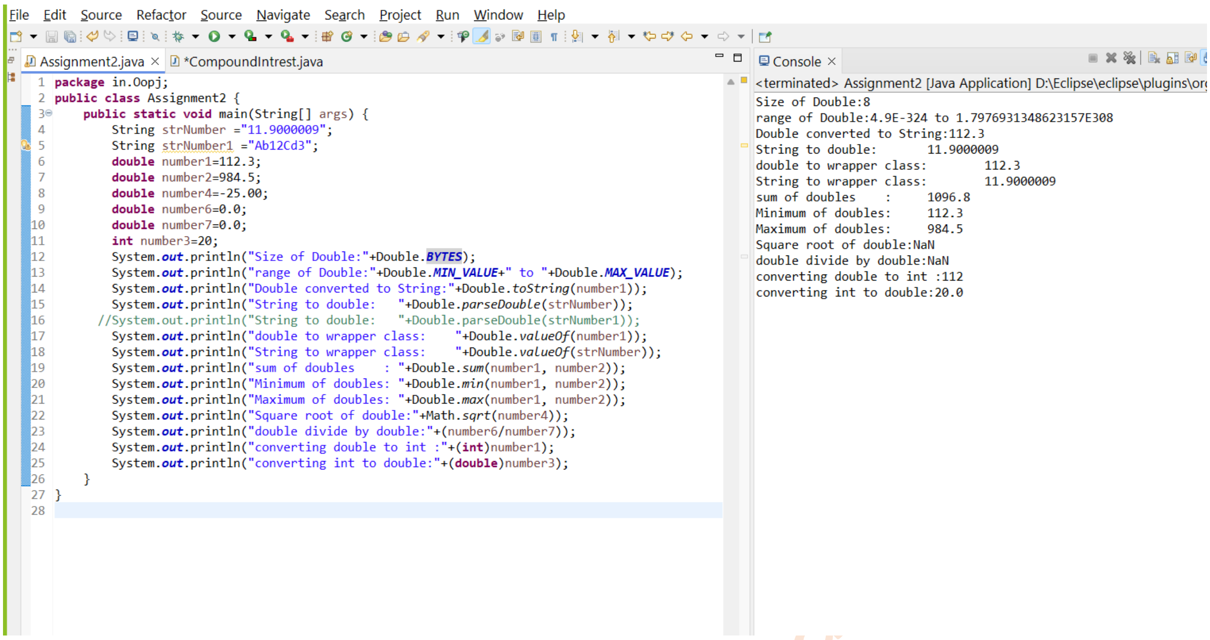
k. Declare a double variable with the value `-25.0`. Find the square root of this value. (Hint: Use `Math.sqrt()` method).

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

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

```
package Shape;import java.util.Scanner;

public class Area {

	public static void main(String[] args) {
     int x;do {

    System.out.println("1.Circle \n2.Square \n3.Rectangle \n4.Triangle");

    System.out.println("Enter your Choise(1-4): ");

    Scanner sc =new Scanner (System.in);

    int ch=sc.nextInt();

    switch (ch) {

    case 1: {

        float A=0;

        float pi=3.14f;

        System.out.println("Enter redius of Circle: ");

        float r=sc.nextFloat();

        A=pi*r*r;

        System.out.println("Area of circle is: "+A);

}break;

    case 2: {

        float A=0;

        System.out.println("Enter Side of Square: ");

        float s=sc.nextFloat();

        A=s*s;

        System.out.println("Area of Square: "+A);

    }break;

    case 3:{

        float A=0;

        System.out.println("Enter Breadth of rectangle: ");
```

```
                        float b=sc.nextFloat();

                        System.out.println("Enter Length of rectangle: ");

                        float l=sc.nextFloat();

                        A=l*b;

                        System.out.println("Area of Rectangle: "+A);

                }break;case 4: {

                        float A=0;

                        System.out.println("Enter Base of Triangle: ");

                        float b=sc.nextFloat();

                        System.out.println("Enter Height of Triangle: ");

                        float h=sc.nextFloat();

                        A=1/2f*(b*h);

                        System.out.println("Area of Triangle: "+A);

                }}System.out.println("Do you want to continue \n1.Yes\n2.No");

                 x=sc.nextInt();

                }while (x==1);}


        }
```


File Edit Source Refactor Source Navigate Search Project Run Window Help
Assignment2.java
*CompoundIntrest.java
Console

```java
package in.Oopj;
public class Assignment2 {
    public static void main(String[] args) {
        String strNumber ="11.9000009";
        String strNumber1 ="Ab12Cd3";
        double number1=112.3;
        double number2=984.5;
        double number4=-25.00;
        double number6=0.0;
        double number7=0.0;
        int number3=20;
        System.out.println("Size of Double:"+Double.BYTES);
        System.out.println("range of Double:"+Double.MIN_VALUE+" to "+Double.MAX_VALUE);
        System.out.println("Double converted to String:"+Double.toString(number1));
        System.out.println("String to double:   "+Double.parseDouble(strNumber));
      //System.out.println("String to double:   "+Double.parseDouble(strNumber1));
        System.out.println("double to wrapper class:    "+Double.valueOf(number1));
        System.out.println("String to wrapper class:    "+Double.valueOf(strNumber));
        System.out.println("sum of doubles    : "+Double.sum(number1, number2));
        System.out.println("Minimum of doubles: "+Double.min(number1, number2));
        System.out.println("Maximum of doubles: "+Double.max(number1, number2));
        System.out.println("Square root of double:"+Math.sqrt(number4));
        System.out.println("double divide by double:"+(number6/number7));
        System.out.println("converting double to int :"+(int)number1);
        System.out.println("converting int to double:"+(double)number3);
    }
}
```

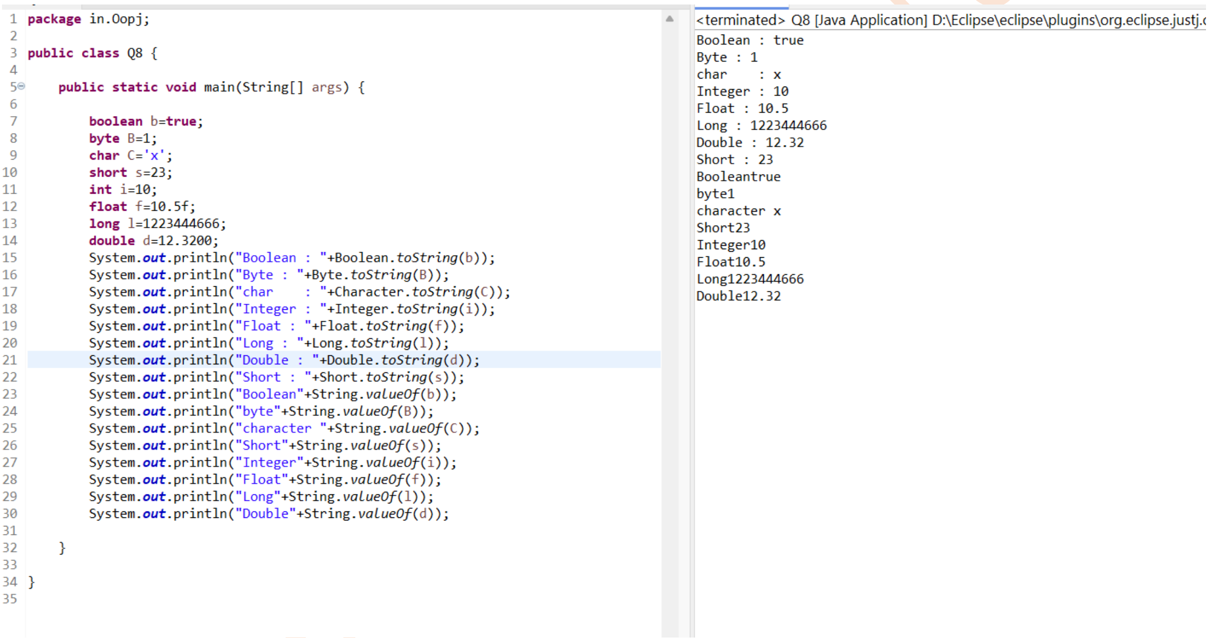
```
<terminated> Assignment2 [Java Application] D:\Eclipse\eclipse\plugins\org
Size of Double:8
range of Double:4.9E-324 to 1.7976931348623157E308
Double converted to String:112.3
String to double:       11.9000009
double to wrapper class:        112.3
String to wrapper class:        11.9000009
sum of doubles    :     1096.8
Minimum of doubles:     112.3
Maximum of doubles:     984.5
Square root of double:NaN
double divide by double:NaN
converting double to int :112
converting int to double:20.0
```

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()`).

```java
package in.Oopj;

public class Q8 {

    public static void main(String[] args) {

            boolean b=true;
            byte B=1;
            char C='x';
            short s=23;
            int i=10;
            float f=10.5f;
            long l=1223444666;
            double d=12.3200;
            System.out.println("Boolean : "+Boolean.toString(b));
            System.out.println("Byte : "+Byte.toString(B));
            System.out.println("char    : "+Character.toString(C));
            System.out.println("Integer : "+Integer.toString(i));
            System.out.println("Float : "+Float.toString(f));
            System.out.println("Long : "+Long.toString(l));
            System.out.println("Double : "+Double.toString(d));
            System.out.println("Short : "+Short.toString(s));
            System.out.println("Boolean"+String.valueOf(b));
            System.out.println("byte"+String.valueOf(B));
            System.out.println("character  "+String.valueOf(C));
            System.out.println("Short"+String.valueOf(s));
            System.out.println("Integer"+String.valueOf(i));
            System.out.println("Float"+String.valueOf(f));
            System.out.println("Long"+String.valueOf(l));
            System.out.println("Double"+String.valueOf(d));

    }

}
```



```
package in.Oopj;

public class Q8 {

    public static void main(String[] args) {

        boolean b=true;
        byte B=1;
        char C='x';
        short s=23;
        int i=10;
        float f=10.5f;
        long l=1223444666;
        double d=12.3200;
        System.out.println("Boolean : "+Boolean.toString(b));
        System.out.println("Byte : "+Byte.toString(B));
        System.out.println("char    : "+Character.toString(C));
        System.out.println("Integer : "+Integer.toString(i));
        System.out.println("Float : "+Float.toString(f));
        System.out.println("Long : "+Long.toString(l));
        System.out.println("Double : "+Double.toString(d));
        System.out.println("Short : "+Short.toString(s));
        System.out.println("Boolean"+String.valueOf(b));
        System.out.println("byte"+String.valueOf(B));
        System.out.println("character "+String.valueOf(C));
        System.out.println("Short"+String.valueOf(s));
        System.out.println("Integer"+String.valueOf(i));
        System.out.println("Float"+String.valueOf(f));
        System.out.println("Long"+String.valueOf(l));
        System.out.println("Double"+String.valueOf(d));

    }

}
```

<terminated> Q8 [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.

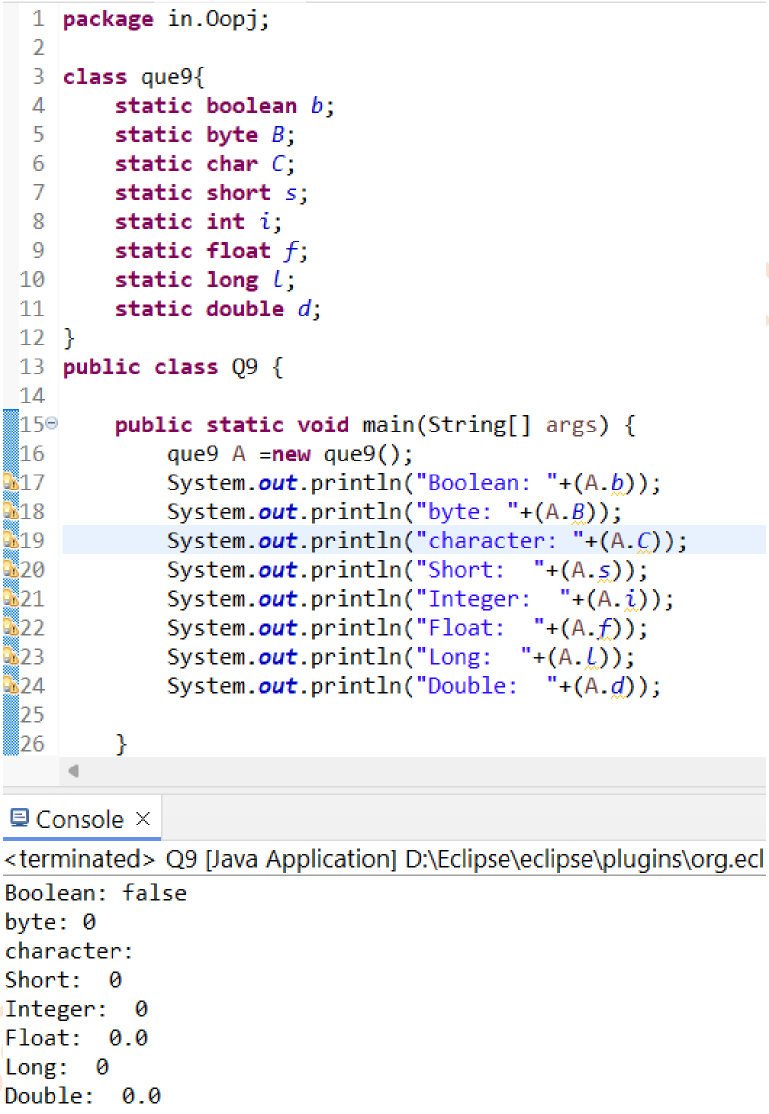
```
Boolean : true
Byte : 1
char    : x
Integer : 10
Float : 10.5
Long : 1223444666
Double : 12.32
Short : 23
Booleantrue
byte1
character x
Short23
Integer10
Float10.5
Long1223444666
Double12.32
```

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).

```
package in.Oopj

class que9{

        static boolean b;

        static byte B;

        static char C;

        static short s;

        static int i;

        static float f;

        static long l;

        static double d;

}

public class Q9 {

        public static void main(String[] args) {

                que9 A =new que9();

                System.out.println("Boolean: "+(A.b));

                System.out.println("byte: "+(A.B));

                System.out.println("character: "+(A.C));

                System.out.println("Short:  "+(A.s));

                System.out.println("Integer:  "+(A.i));

                System.out.println("Float:  "+(A.f));
```

```
                    System.out.println("Long:  "+(A.l));

                    System.out.println("Double:  "+(A.d));

          }

}
```



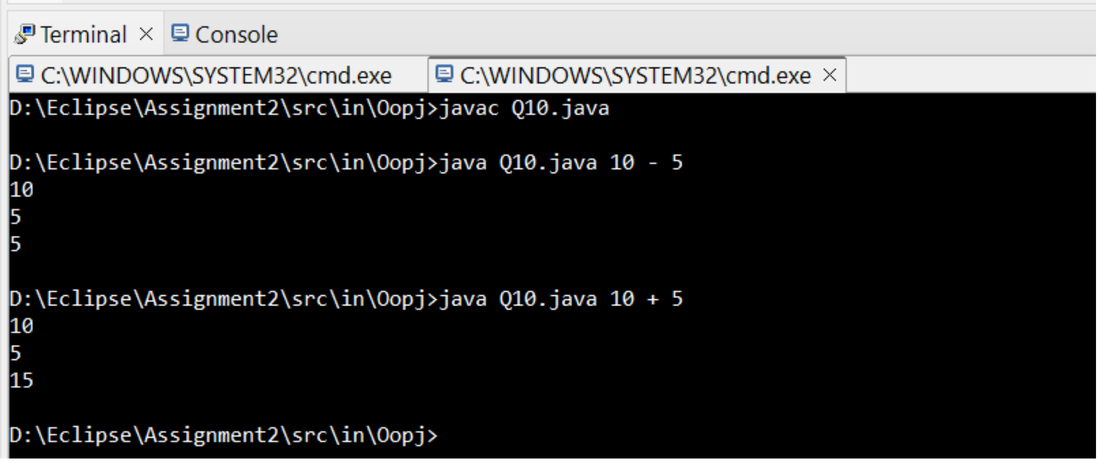
```
package in.Oopj;

class que9{
    static boolean b;
    static byte B;
    static char C;
    static short s;
    static int i;
    static float f;
    static long l;
    static double d;
}
public class Q9 {

    public static void main(String[] args) {
        que9 A =new que9();
        System.out.println("Boolean: "+(A.b));
        System.out.println("byte: "+(A.B));
        System.out.println("character: "+(A.C));
        System.out.println("Short:  "+(A.s));
        System.out.println("Integer:  "+(A.i));
        System.out.println("Float:  "+(A.f));
        System.out.println("Long:  "+(A.l));
        System.out.println("Double:  "+(A.d));

    }
```

```
<terminated> Q9 [Java Application] D:\Eclipse\eclipse\plugins\org.ecl
Boolean: false
byte: 0
character: 
Short:  0
Integer:  0
Float:  0.0
Long:  0
Double:  0.0
```

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).

```
package in.Oopj;

public class Q10 {
                    public static void main(String[] args) {
                              int num1 = Integer.parseInt(args[0]);
                       System.out.println(num1);
                       String opr = args[1];
                       int num2 = Integer.parseInt(args[2]);
                    System.out.println(num2);

                    switch (opr) {
                       case "+":
                       System.out.println(num1+num2);

                          break;
                    case "-":
                    System.out.println(num1-num2);
                    break;

                    case "*":
                    System.out.println(num1*num2);
                    break;
                    case "/":
                    System.out.println(num1/num2);
                       default:
                          break;
                    }
          }
}
```



```
D:\Eclipse\Assignment2\src\in\Oopj>javac Q10.java

D:\Eclipse\Assignment2\src\in\Oopj>java Q10.java 10 - 5
10
5
5

D:\Eclipse\Assignment2\src\in\Oopj>java Q10.java 10 + 5
10
5
15

D:\Eclipse\Assignment2\src\in\Oopj>
```