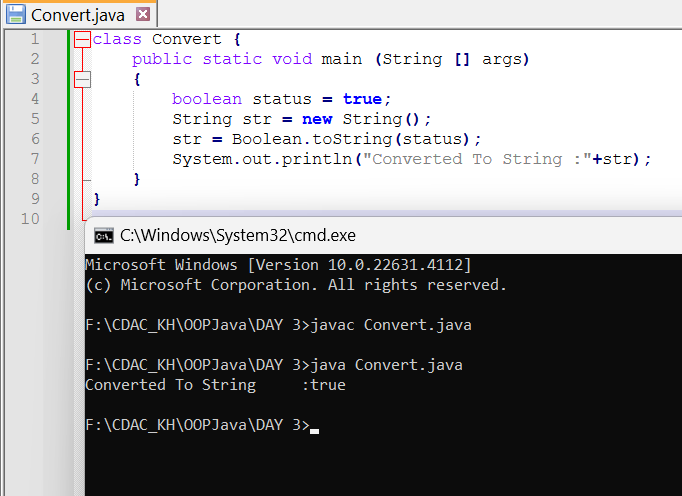
**Note**: Consider the following before starting the assignment:

* 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.

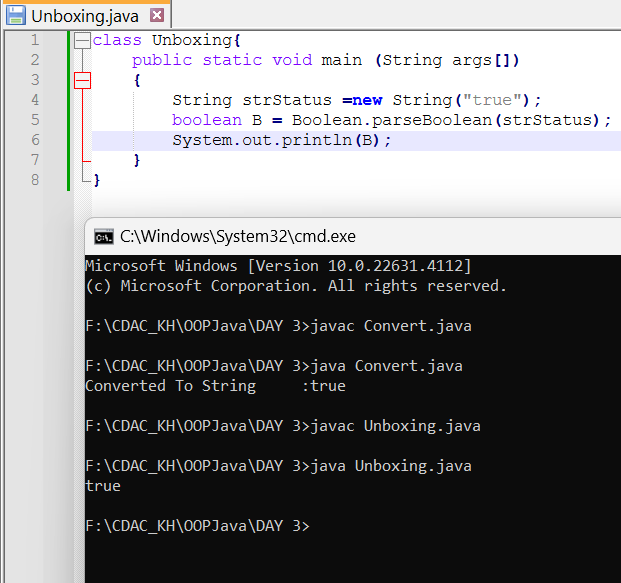
#### ****1. Working with**** java.lang.Boolean

**a.** Explore the [Java API documentation for java.lang.Boolean](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) and observe its modifiers and super types.

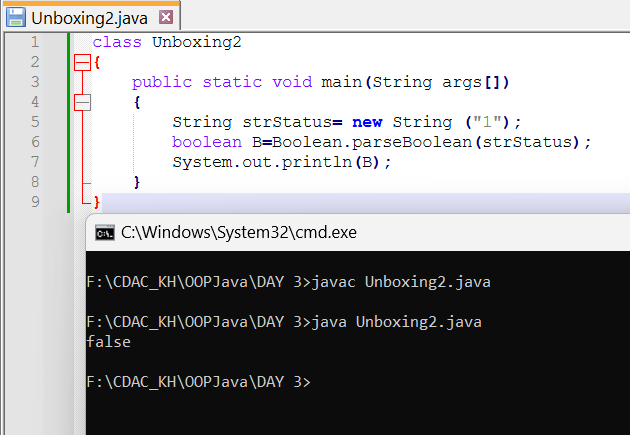
**b.** Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method. (Hint: Use Boolean.toString(Boolean).



**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)).

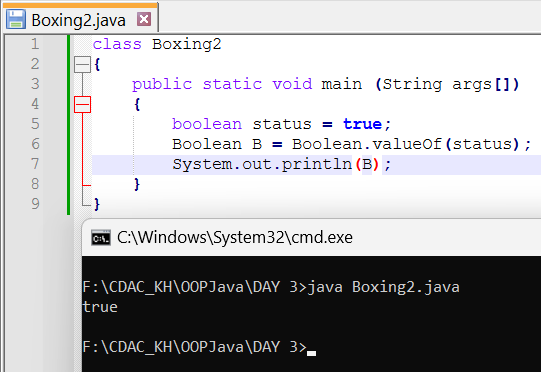


**d.** Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean. (Hint: parseBoolean method will not work as expected with "1" or "0").

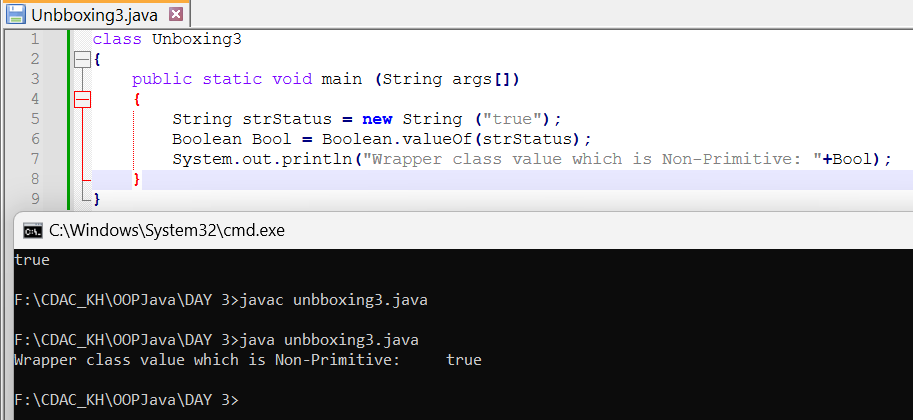


Explanation: As Boolean stores only **true or false** in java but string strStatus passes the value “1” or “0” which result unexpected output.

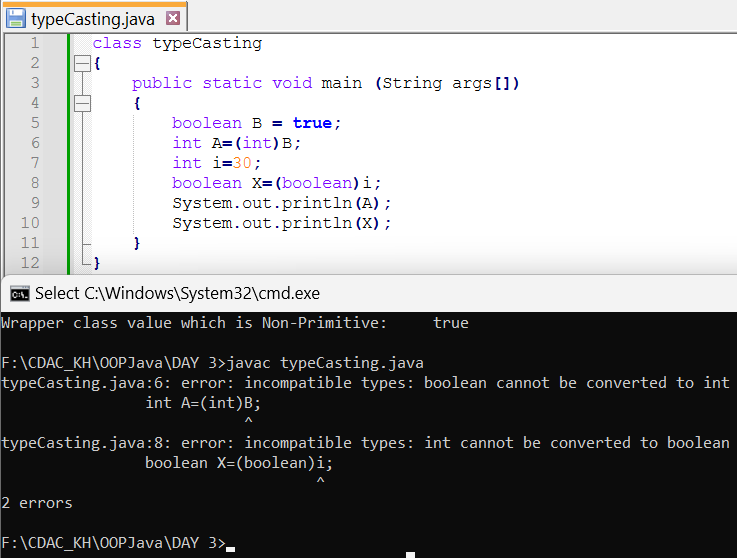
**e.** Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(boolean)).



**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)).



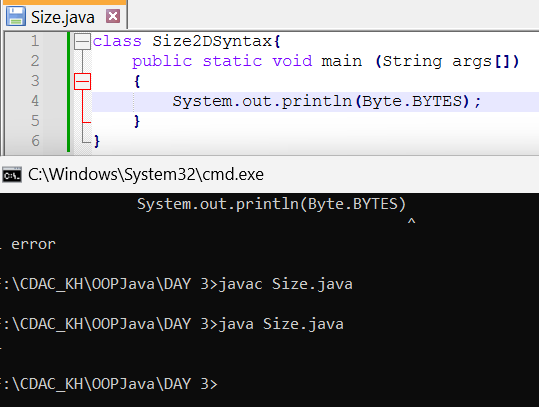
**g.** Experiment with converting a boolean value into other primitive types or vice versa and observe the results.



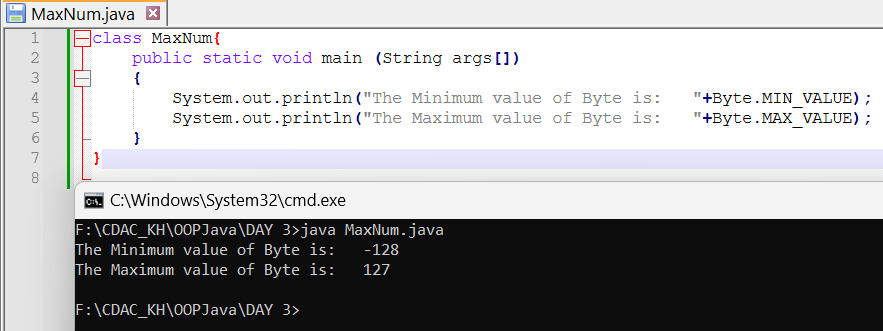
#### ****2. Working with**** java.lang.Byte

**a.** Explore the [Java API documentation for java.lang.Byte](https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html) and observe its modifiers and super types.

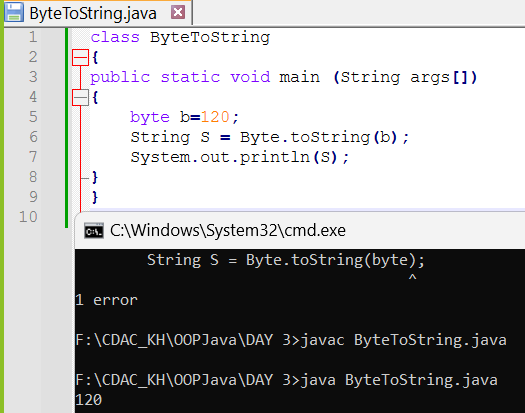
**b.** Write a program to test how many bytes are used to represent a byte value using the BYTES field. (Hint: Use Byte.BYTES).



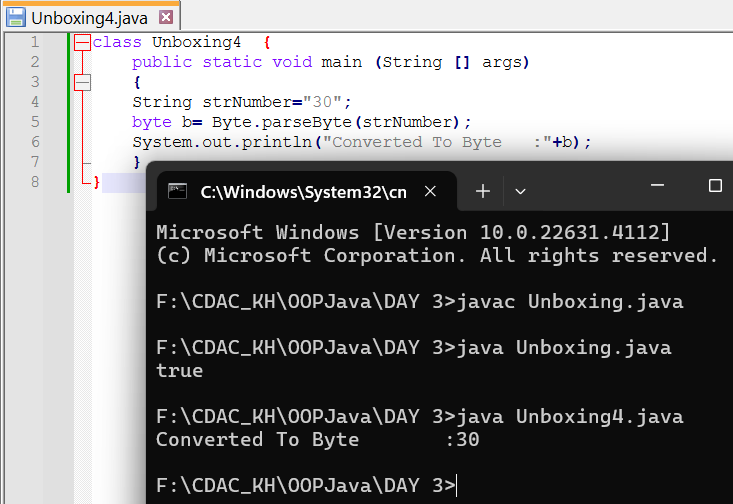
**c.** Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and Byte.MAX\_VALUE).



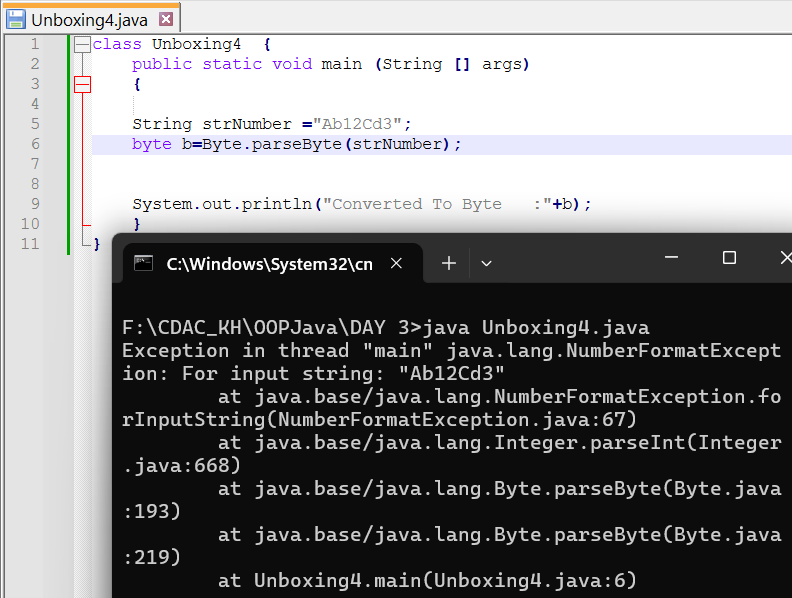
**d.** Declare a method-local variable number of type byte with some value and convert it to a String using the toString method. (Hint: Use Byte.toString(byte)).



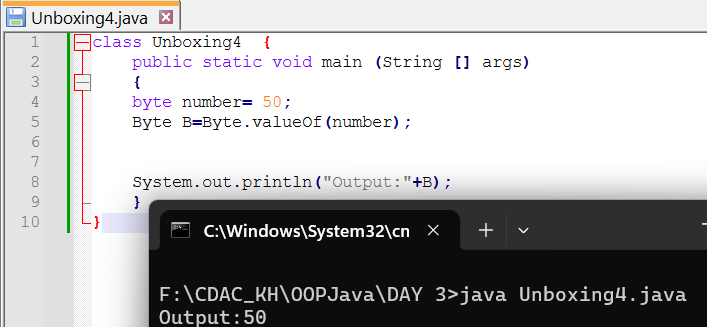
**e.** Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method. (Hint: Use Byte.parseByte(String)).

****

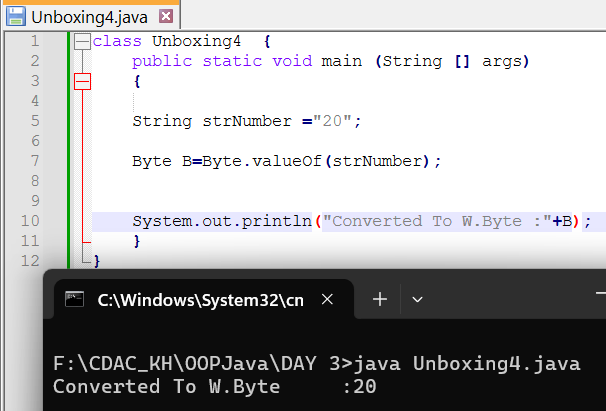
**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value. (Hint: parseByte method will throw a NumberFormatException).



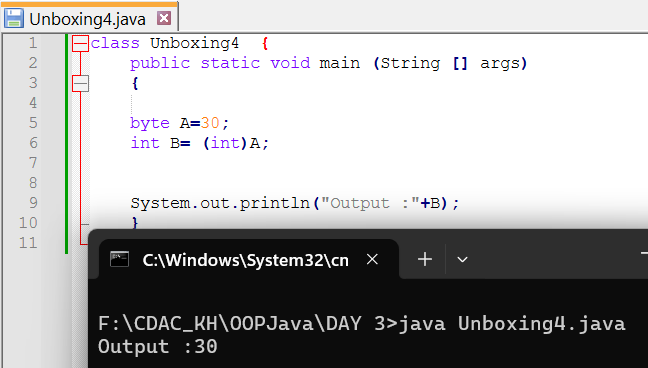
**g.** Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(byte)).



**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)).



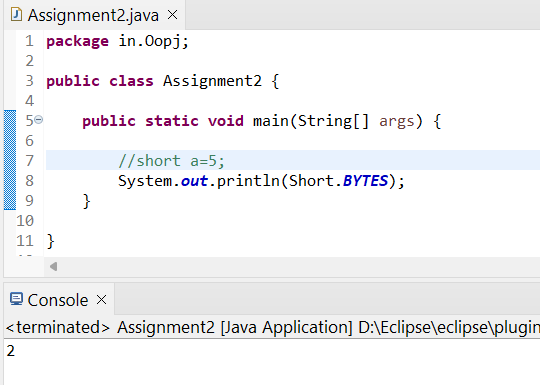
1. Experiment with converting a byte value into other primitive types or vice versa and observe the results.



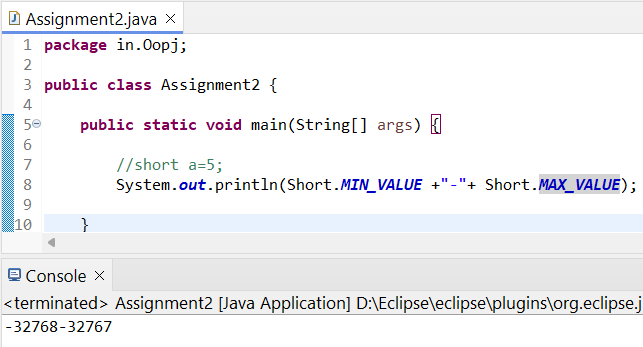
#### ****3. Working with**** java.lang.Short

**a.** Explore the [Java API documentation for java.lang.Short](https://docs.oracle.com/javase/8/docs/api/java/lang/Short.html) and observe its modifiers and super types.

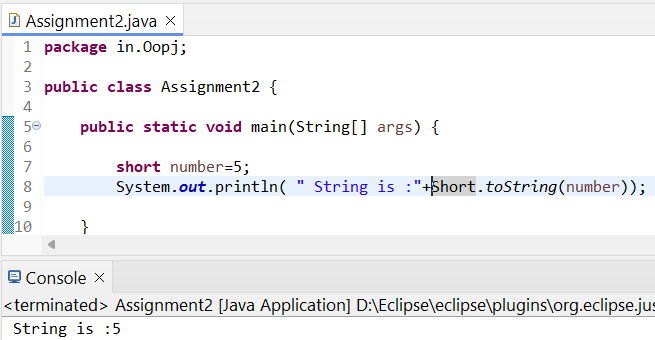
**b.** Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).



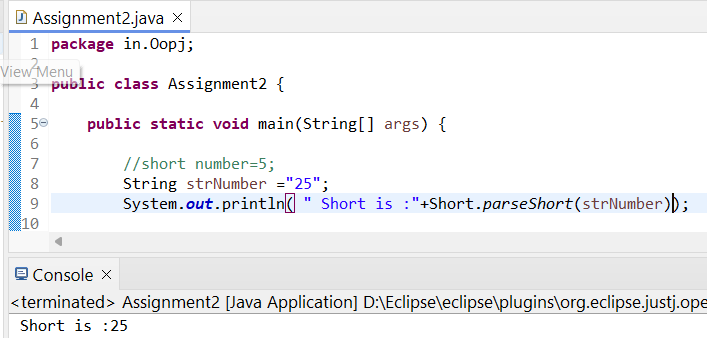
**c.** Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).



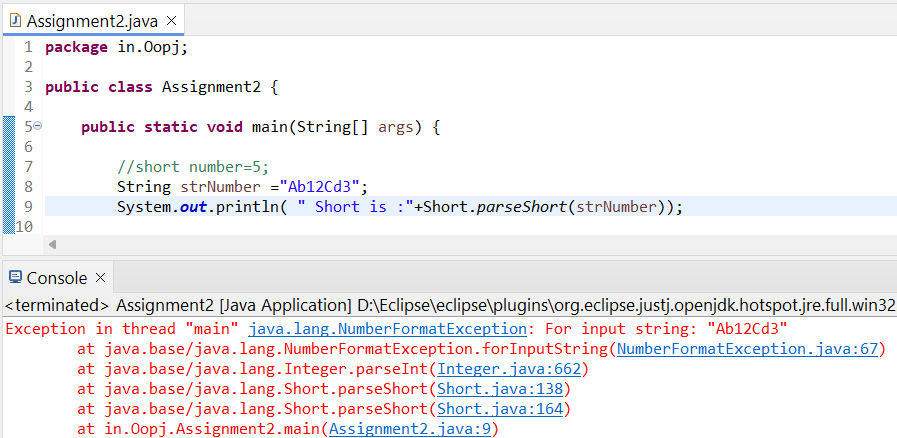
**d.** Declare a method-local variable number of type short with some value and convert it to a String using the toString method. (Hint: Use Short.toString(short)).



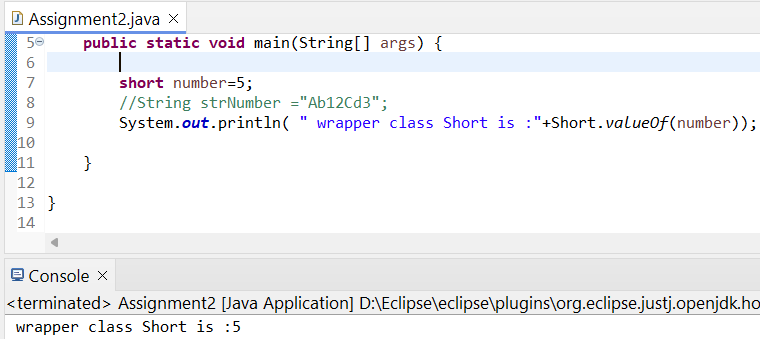
**e.** Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method. (Hint: Use Short.parseShort(String)).



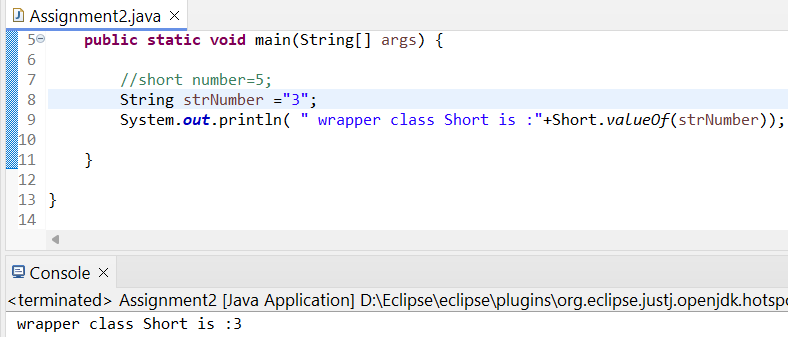
**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value. (Hint: parseShort method will throw a NumberFormatException).



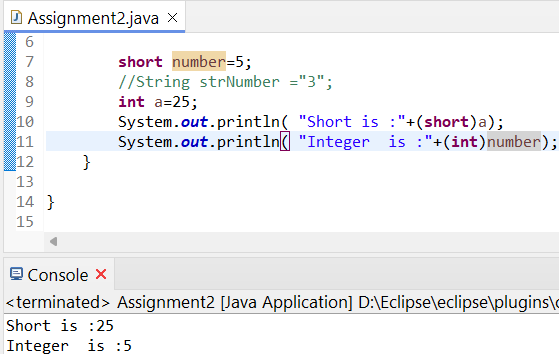
**g.** Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(short)).



**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)).



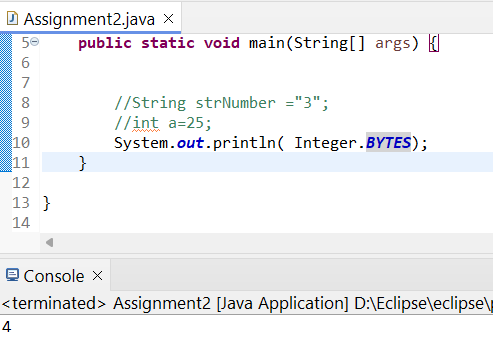
**i.** Experiment with converting a short value into other primitive types or vice versa and observe the results.



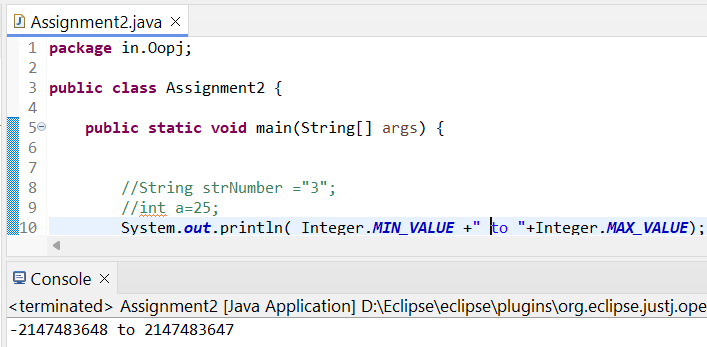
#### ****4. Working with**** java.lang.Integer

**a.** Explore the [Java API documentation for java.lang.Integer](https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html) and observe its modifiers and super types

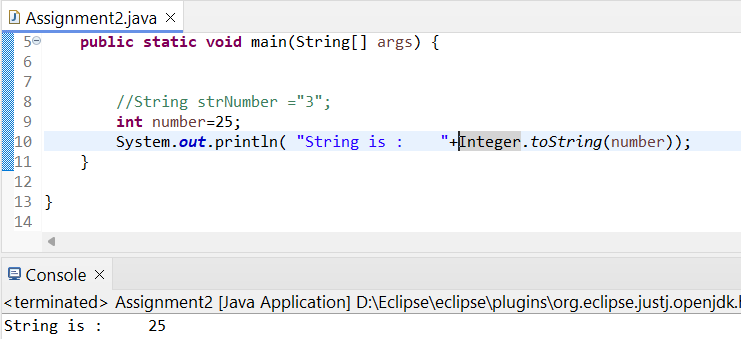
**b.** Write a program to test how many bytes are used to represent an int value using the BYTES field. (Hint: Use Integer.BYTES).



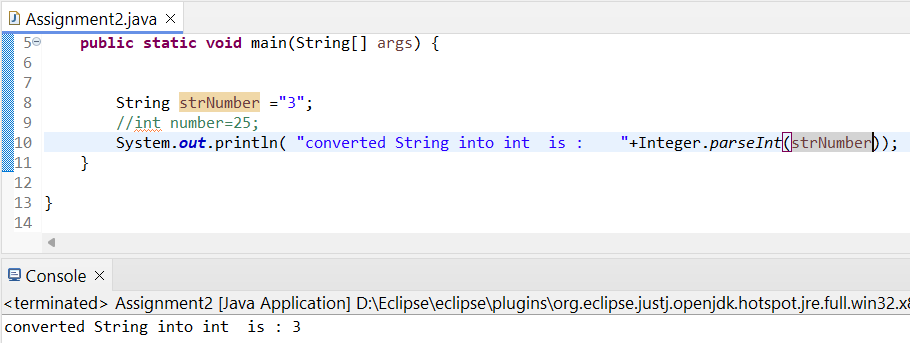
**c.** Write a program to find the minimum and maximum values of int using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Integer.MIN\_VALUE and Integer.MAX\_VALUE).



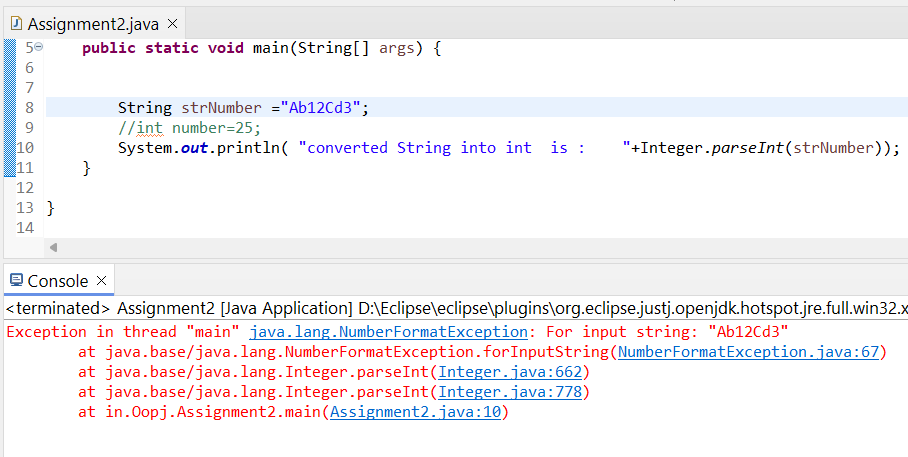
**d.** Declare a method-local variable number of type int with some value and convert it to a String using the toString method. (Hint: Use Integer.toString(int)).



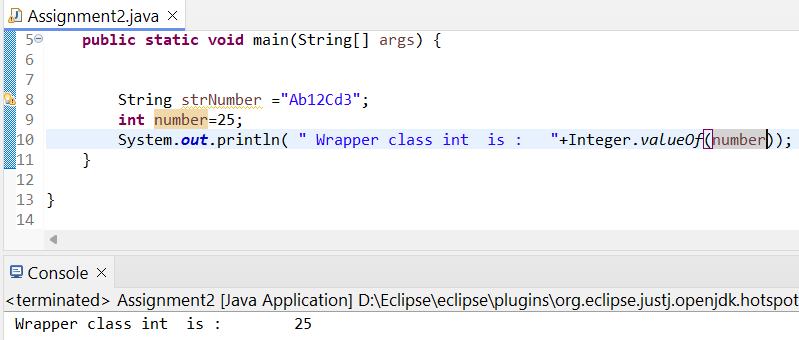
**e.** Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method. (Hint: Use Integer.parseInt(String)).

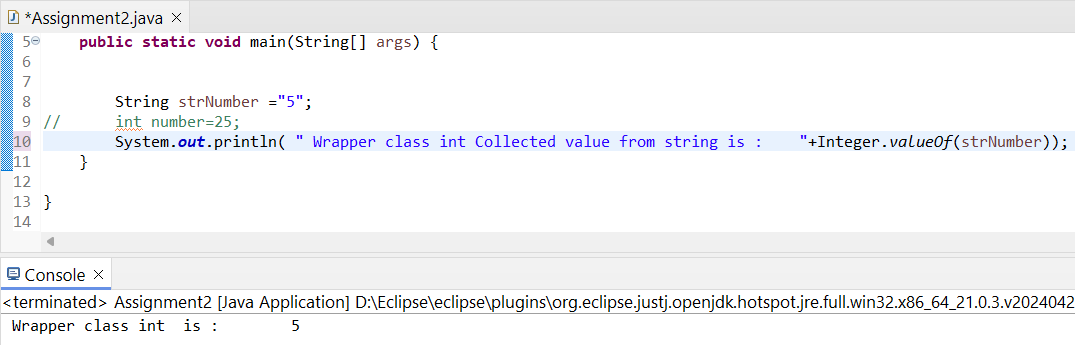


**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value. (Hint: parseInt method will throw a NumberFormatException).

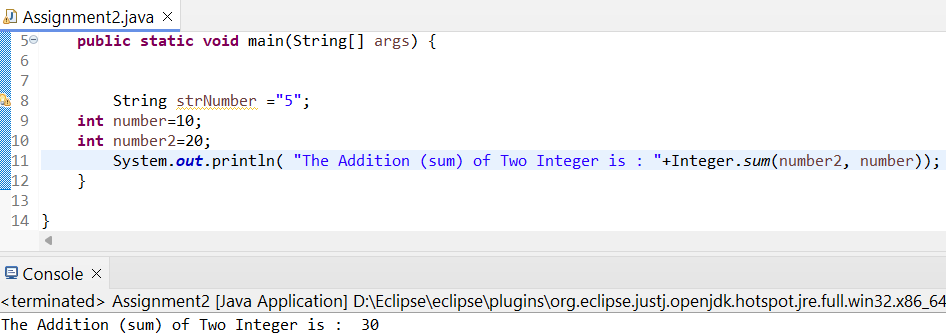


**g.** Declare a method-local variable number of type int with some value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(int)).

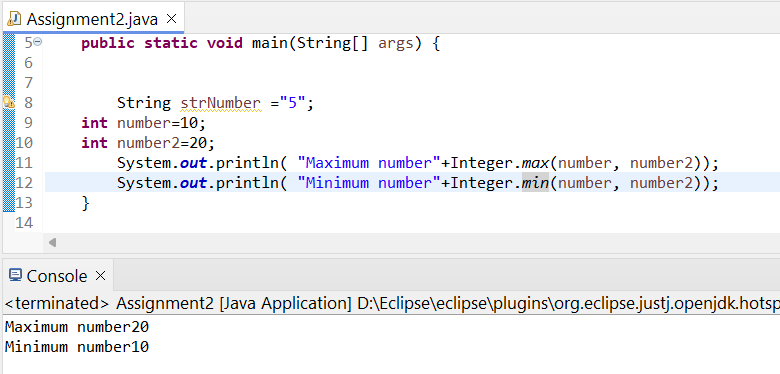


**h.** Declare a method-local variable strNumber of type String with some integer value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(String)). 

**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)).

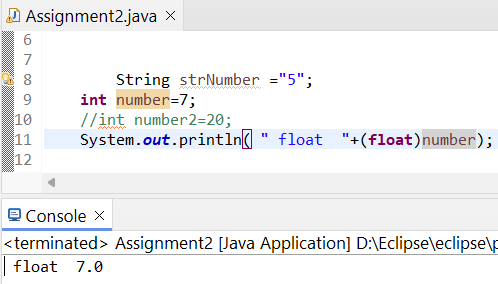


**j.** Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class. (Hint: Use Integer.min(int, int) and Integer.max(int, int)).



**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)).

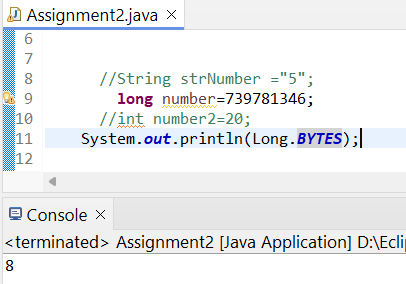


**l.** Experiment with converting an int value into other primitive types or vice versa and observe the results. 

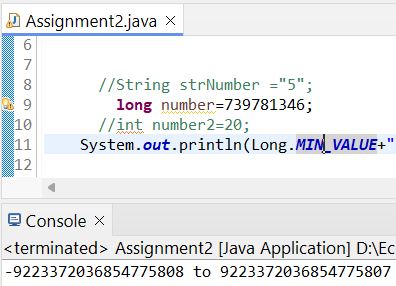
#### ****5. Working with**** java.lang.Long

**a.** Explore the [Java API documentation for java.lang.Long](https://docs.oracle.com/javase/8/docs/api/java/lang/Long.html) and observe its modifiers and super types.

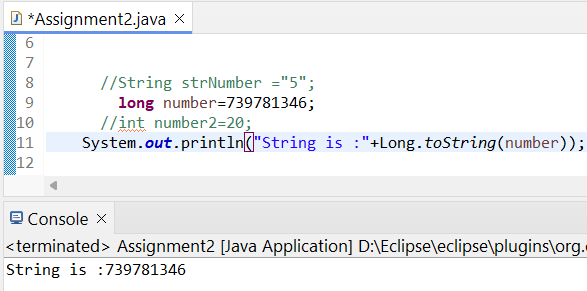
**b.** Write a program to test how many bytes are used to represent a long value using the BYTES field. (Hint: Use Long.BYTES).



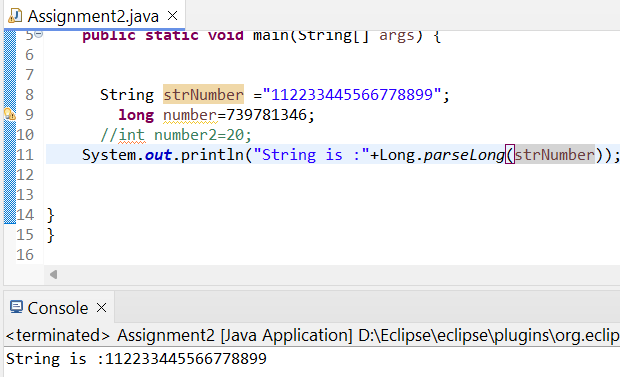
**c.** Write a program to find the minimum and maximum values of long using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Long.MIN\_VALUE and Long.MAX\_VALUE).



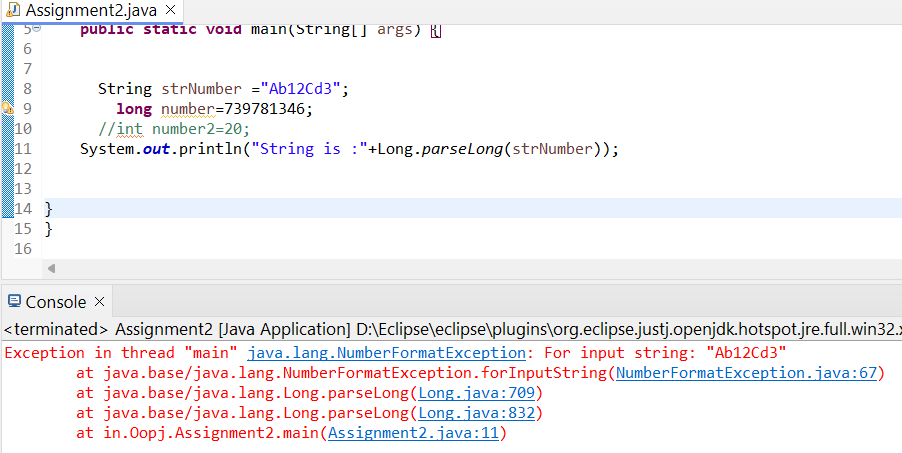
**d.** Declare a method-local variable number of type long with some value and convert it to a String using the toString method. (Hint: Use Long.toString(long)).



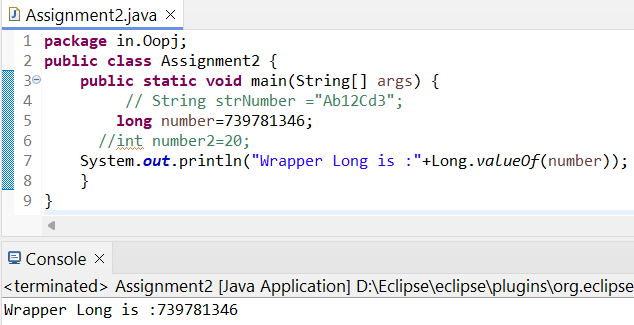
**e.** Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method. (Hint: Use Long.parseLong(String)).



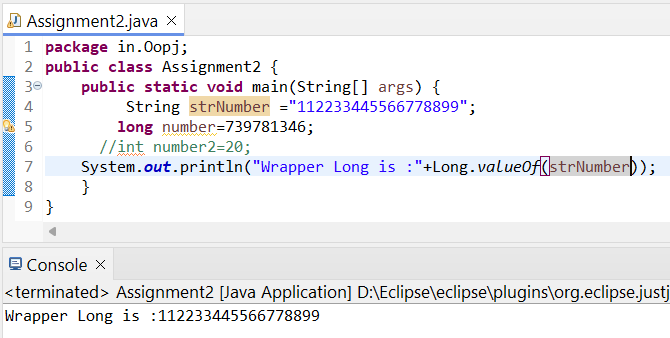
**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value. (Hint: parseLong method will throw a NumberFormatException).



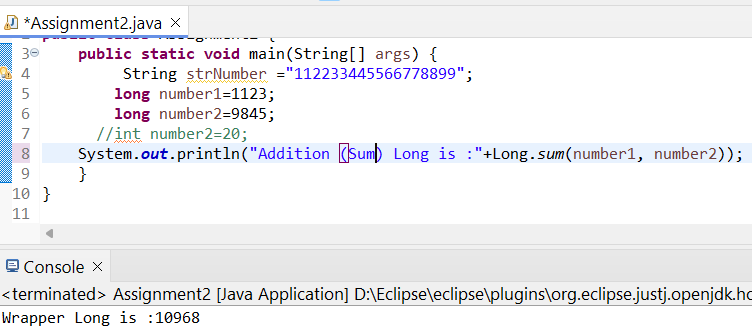
**g.** Declare a method-local variable number of type long with some value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(long)).



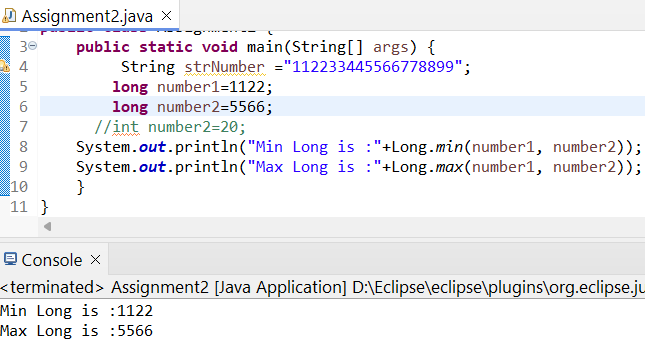
**h.** Declare a method-local variable strNumber of type String with some long value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(String)).



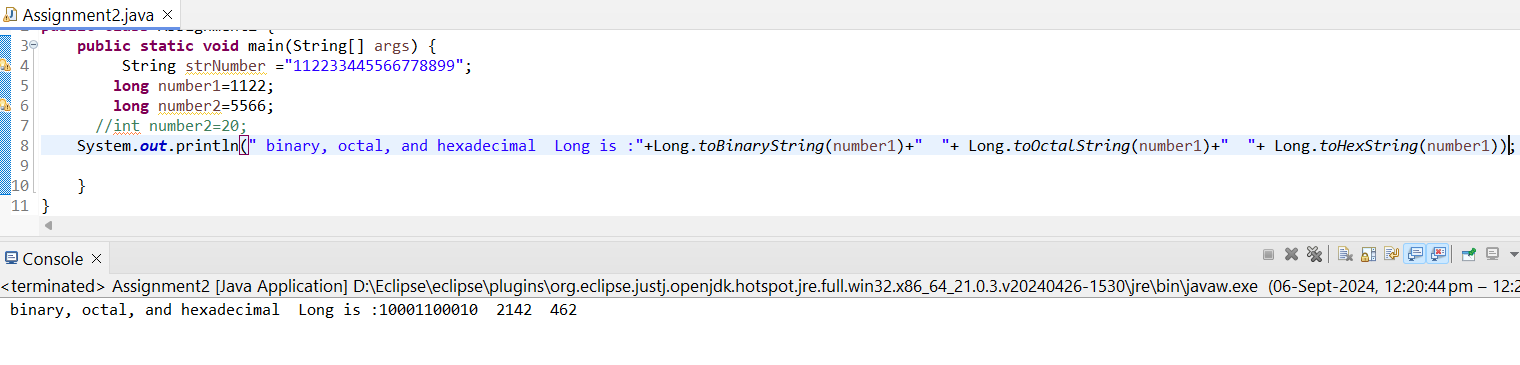
1. Declare two long variables with values 1123 and 9845, and add them using a method from the Long class. (Hint: Use Long.sum(long, long)).



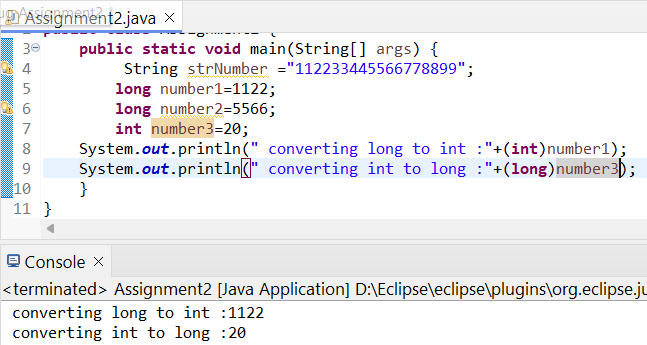
**j.** Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class. (Hint: Use Long.min(long, long) and Long.max(long, long)).



**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)).



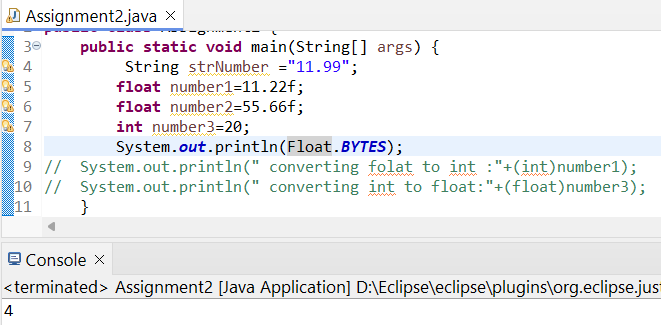
**l.** Experiment with converting a long value into other primitive types or vice versa and observe the results.



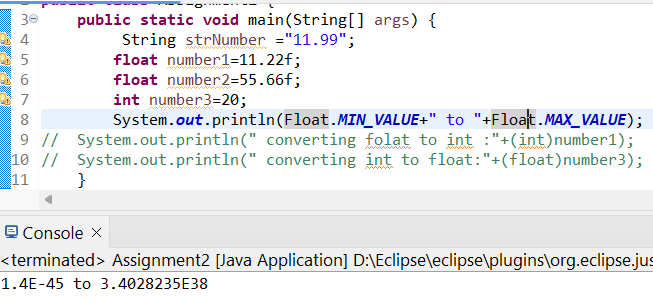
#### ****6. Working with**** java.lang.Float

**a.** Explore the [Java API documentation for java.lang.Float](https://docs.oracle.com/javase/8/docs/api/java/lang/Float.html) and observe its modifiers and super types.

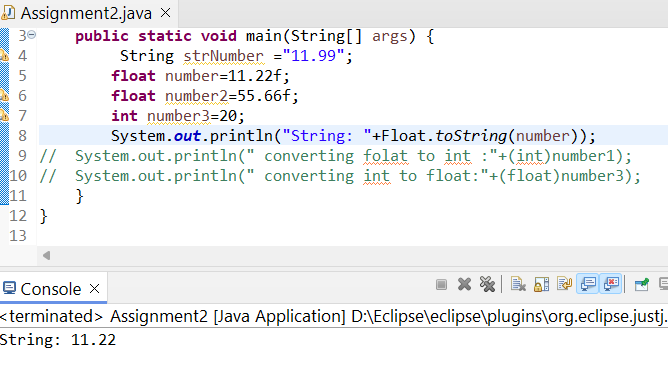
**b.** Write a program to test how many bytes are used to represent a float value using the BYTES field. (Hint: Use Float.BYTES).



**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).

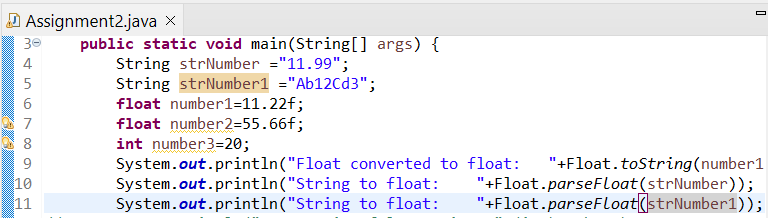


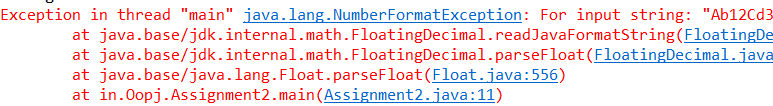
**d.** Declare a method-local variable number of type float with some value and convert it to a String using the toString method. (Hint: Use Float.toString(float)).



**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)).

**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));

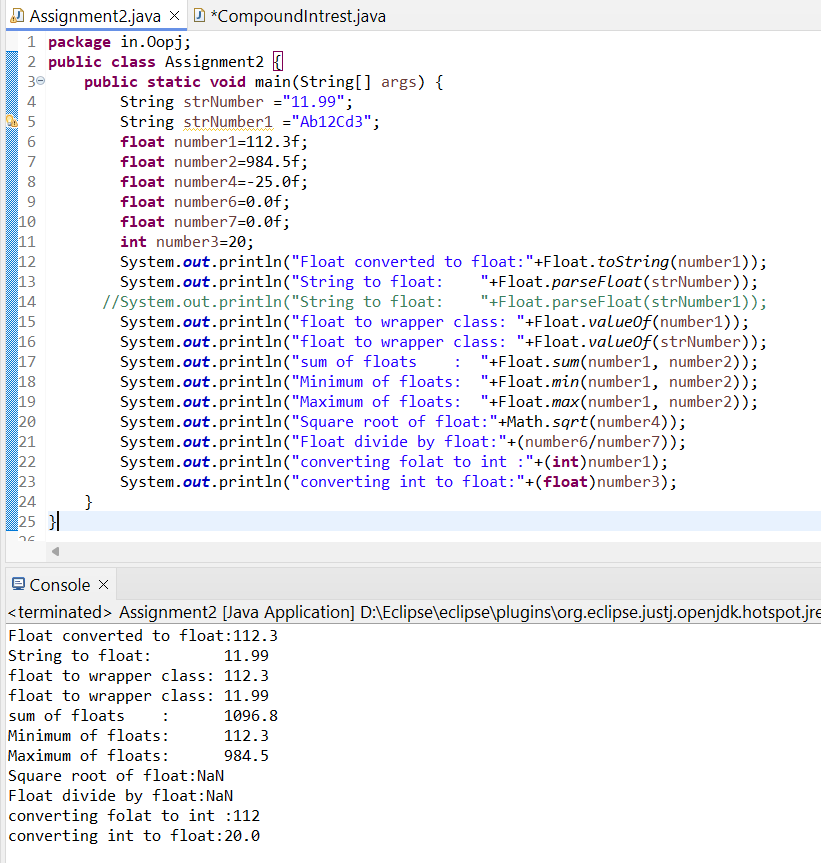
System.***out***.println("Float divide by float:"+(number6/number7));

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

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

}

}



#### ****7. Working with**** java.lang.Double

**a.** Explore the [Java API documentation for java.lang.Double](https://docs.oracle.com/javase/8/docs/api/java/lang/Double.html) and observe its modifiers and super types.

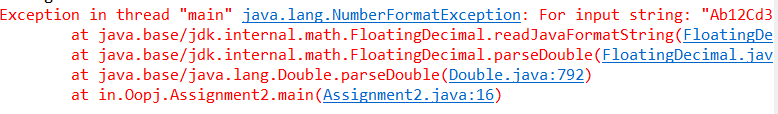
**b.** Write a program to test how many bytes are used to represent a double value using the BYTES field. (Hint: Use Double.BYTES).

**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).



**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);

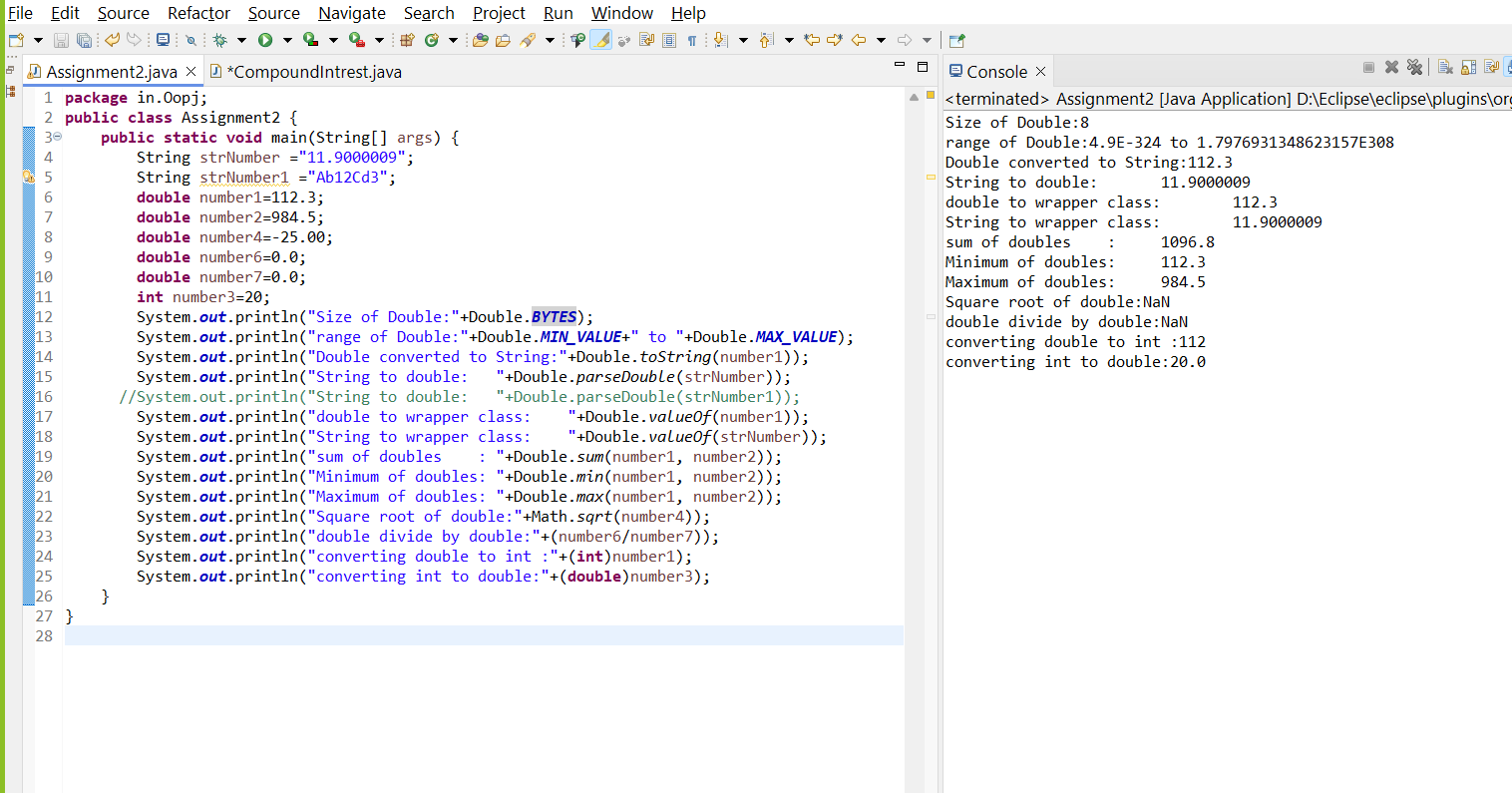
System.***out***.println("Area of Triangle: "+A);

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

x=sc.nextInt();

}**while** (x==1);}

}



#### ****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()).

**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));

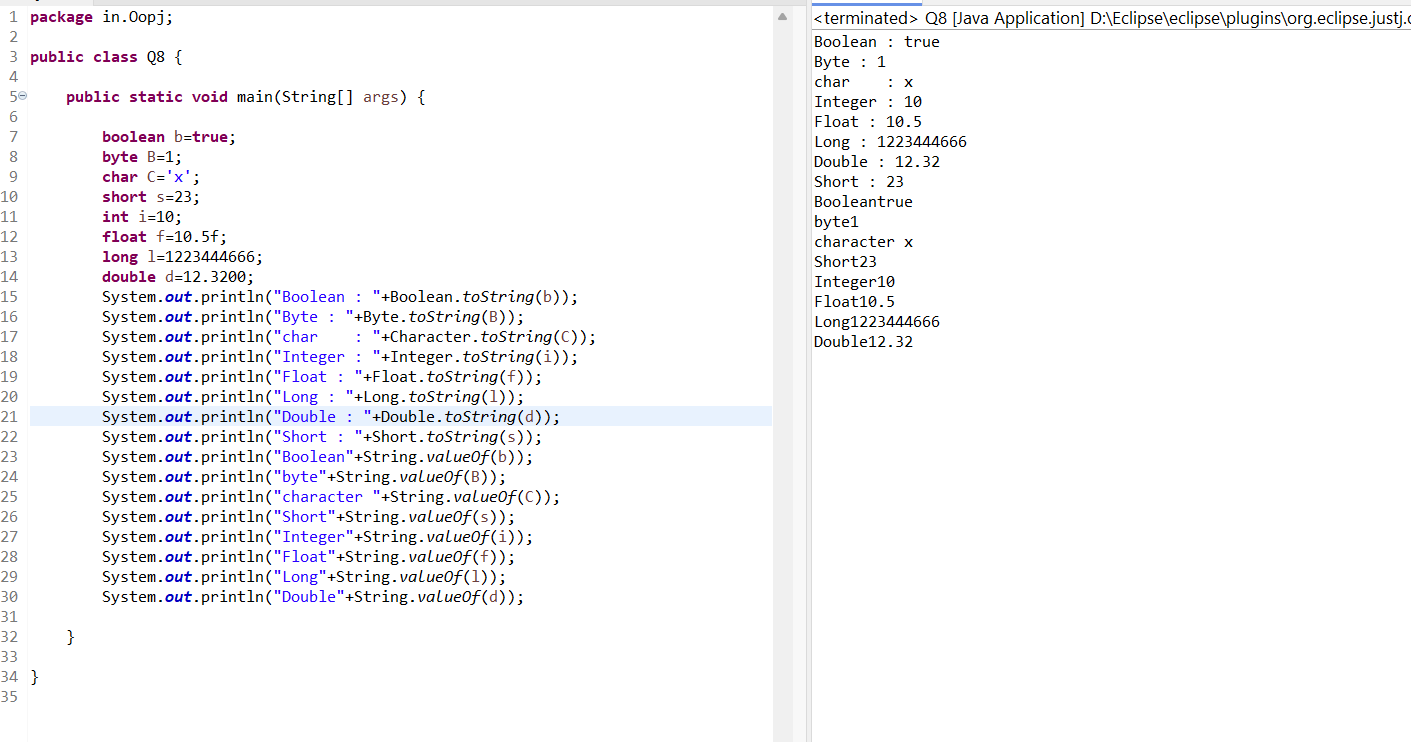
System.***out***.println("Float"+String.*valueOf*(f));

System.***out***.println("Long"+String.*valueOf*(l));

System.***out***.println("Double"+String.*valueOf*(d));

}

}



#### ****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*));

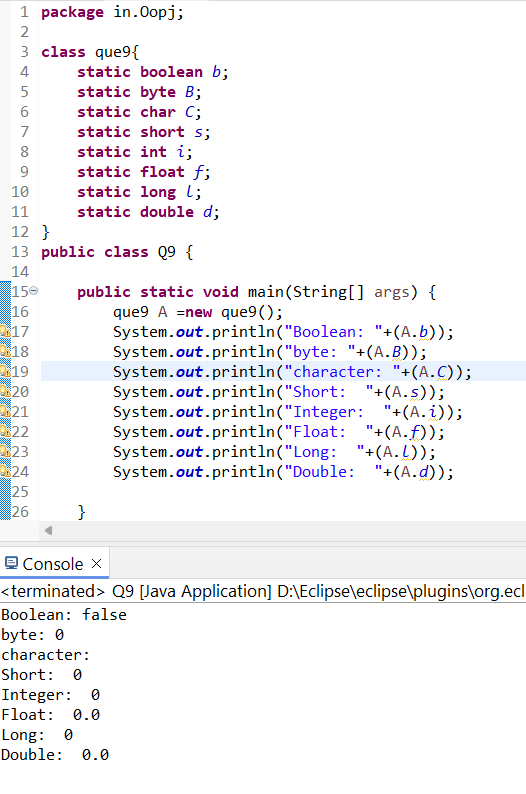
System.***out***.println("Float: "+(A.*f*));

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

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

}

}



#### ****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**;

}

}

}

