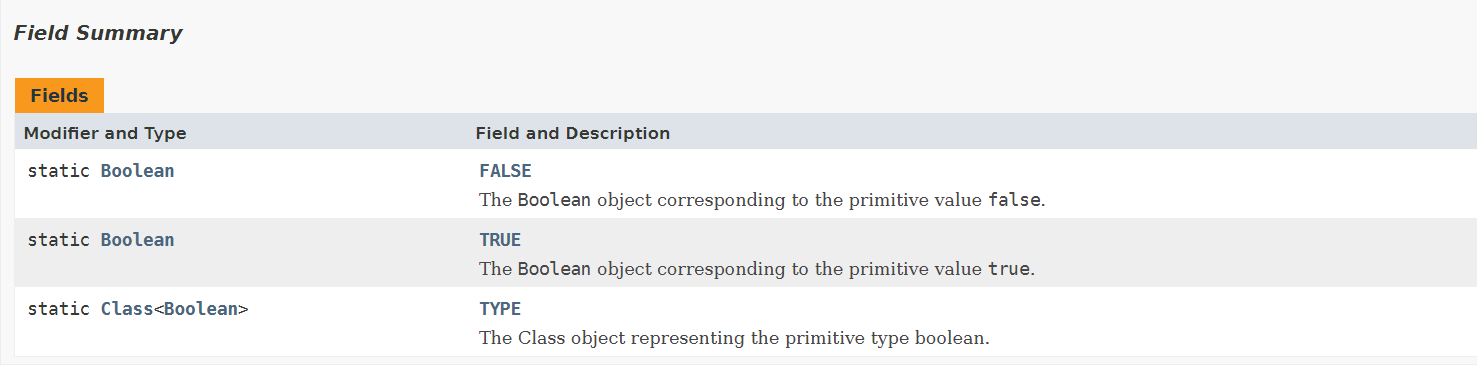
**Note**: Consider the following before starting the assignment:

* 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.

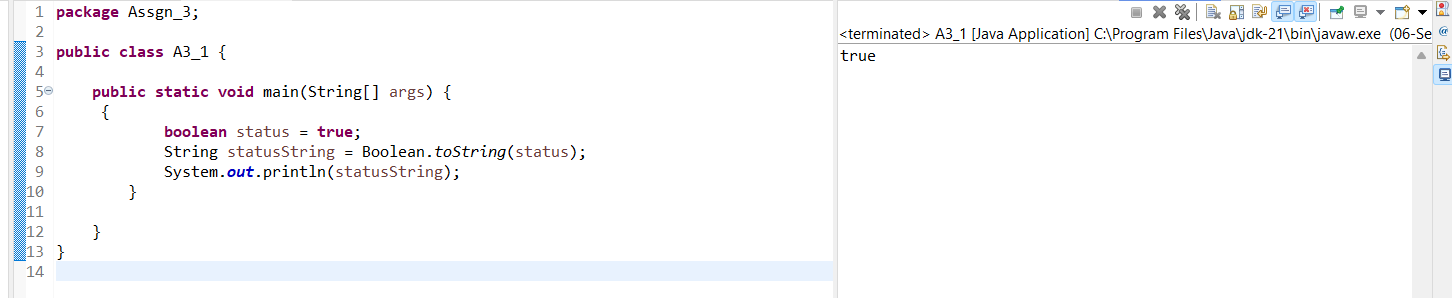
1. Working with **java.lang.Boolean**

**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.

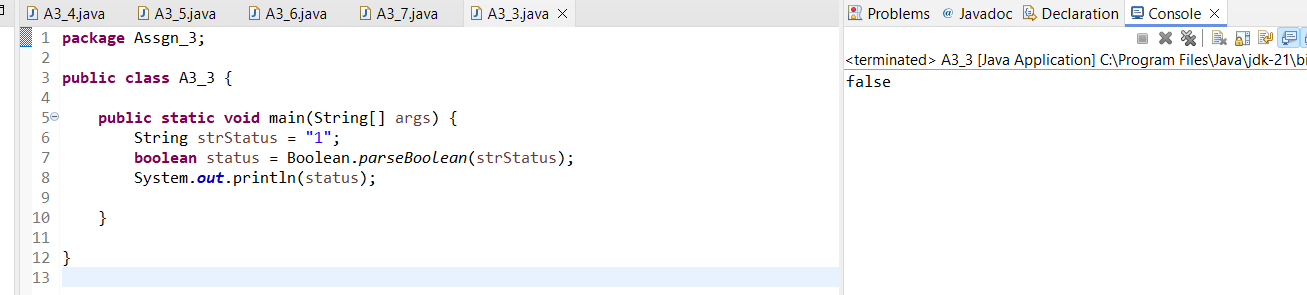
* The Boolean class wraps a value of the primitive type boolean in an object. An object of type Boolean contains a single field whose type is boolean.
* In addition, this class provides many methods for converting a boolean to a String and a String to a boolean, as well as other constants and methods useful when dealing with a boolean.



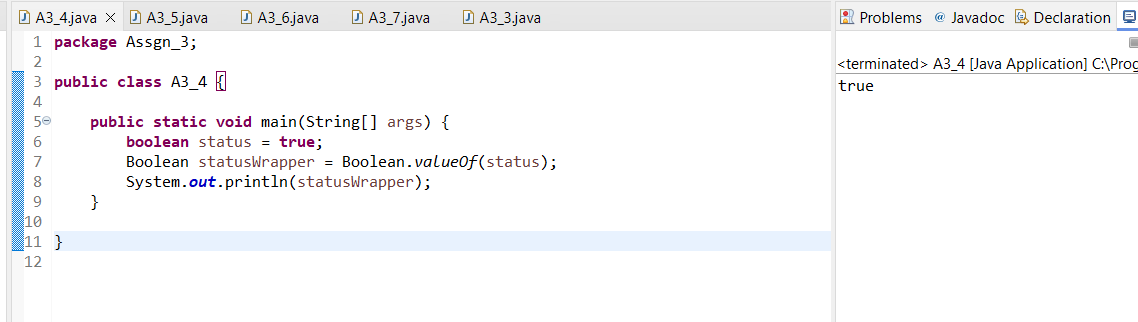
**b.** Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method. (Hint: Use Boolean.toString(Boolean) ).



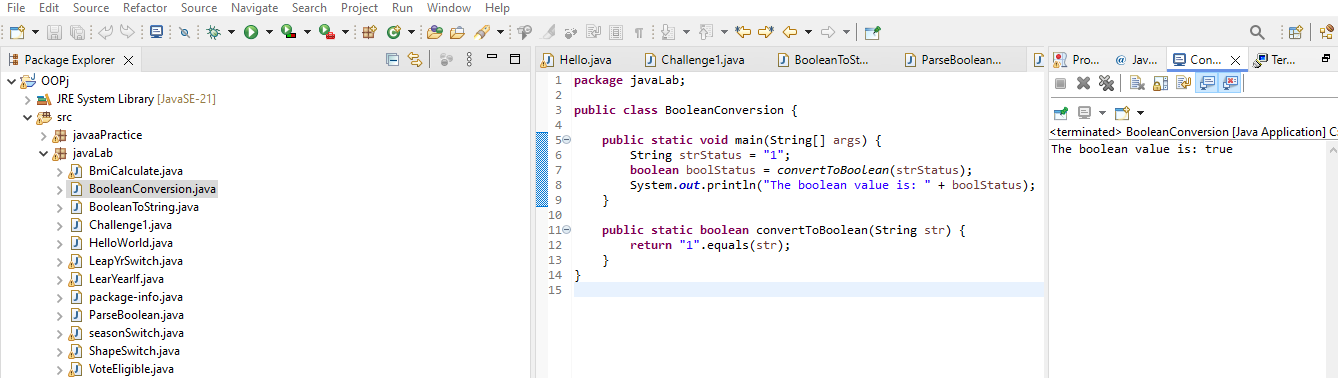
**c.** Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method. (Hint: Use Boolean.parseBoolean(String)).



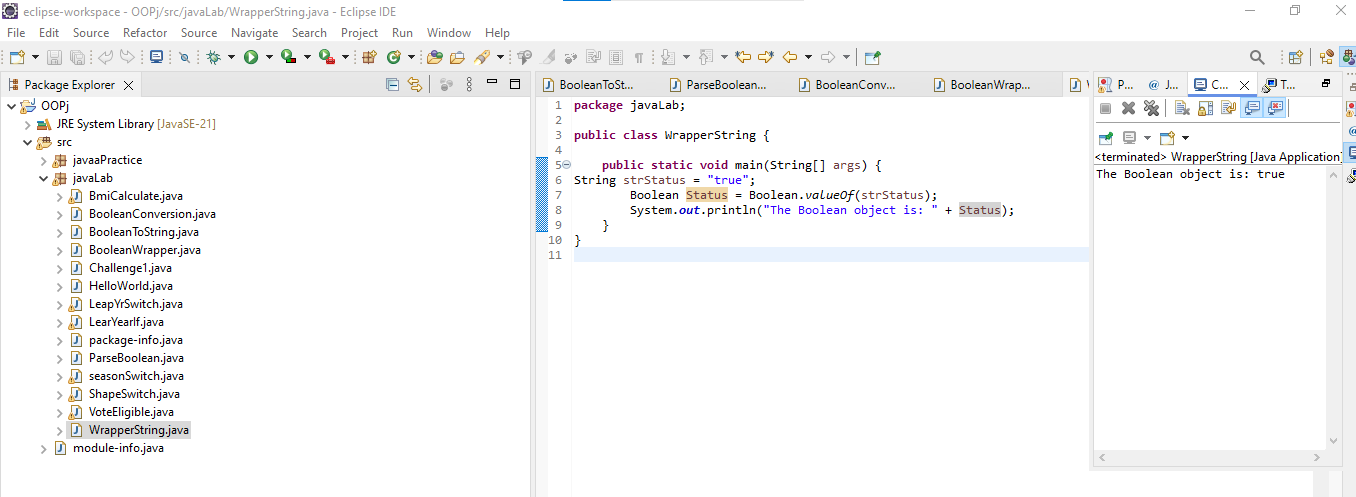
**d.** Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean. (Hint: parseBoolean method will not work as expected with "1" or "0").



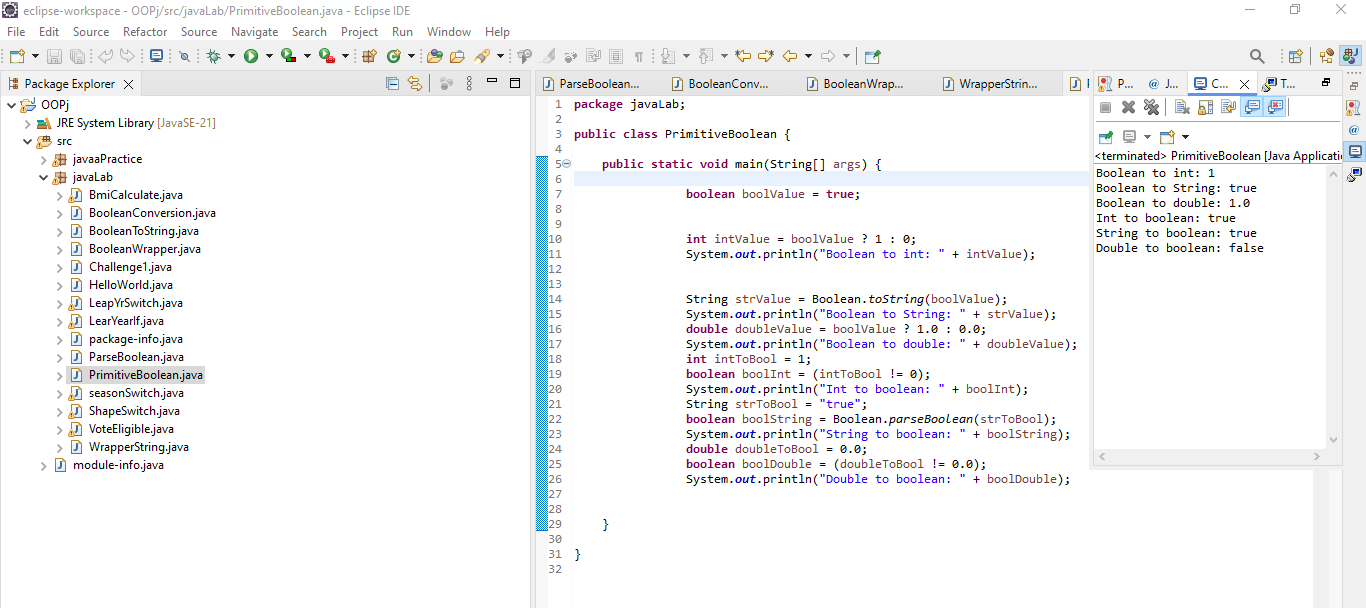
**e.** Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(boolean)).



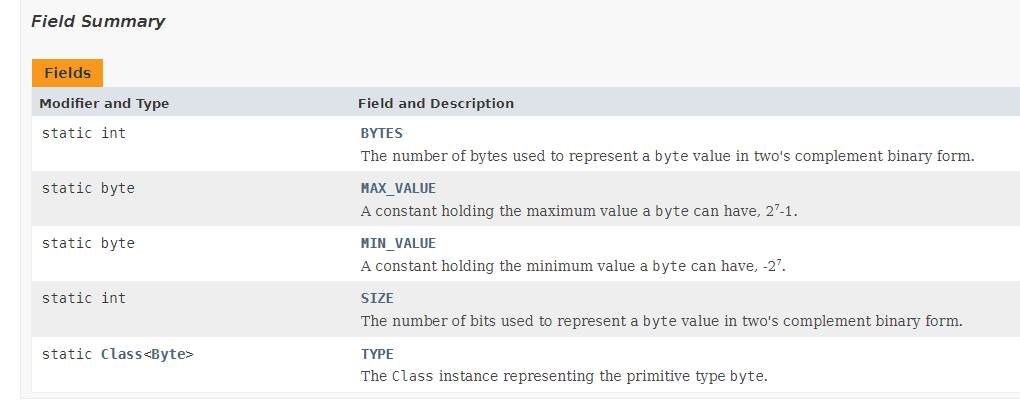
**f.** Declare a method-local variable strStatus of type String with the value "true" and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(String)).

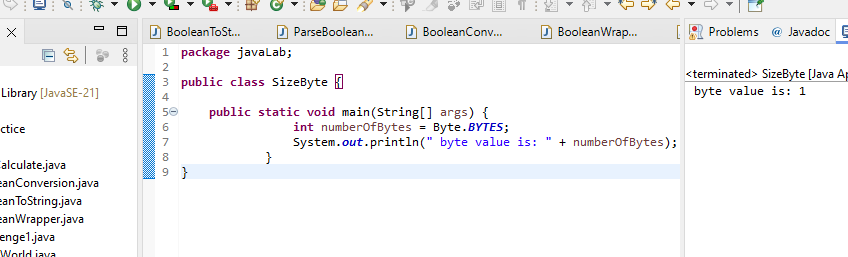


**g.** Experiment with converting a boolean value into other primitive types or vice versa and observe the results.

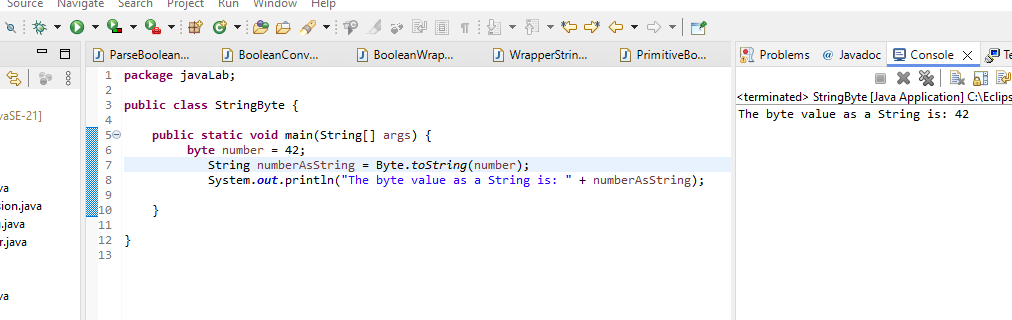


2. Working with **java.lang.Byte**

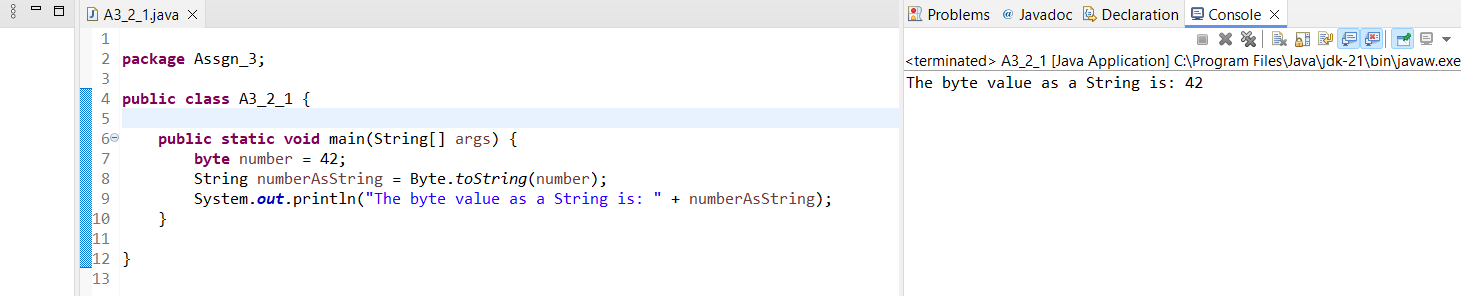
**a.** Explore the [Java API documentation for java.lang.Byte](https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html) and observe its modifiers and super types.**b.** Write a program to test how many bytes are used to represent a byte value using the BYTES field. (Hint: Use Byte.BYTES).



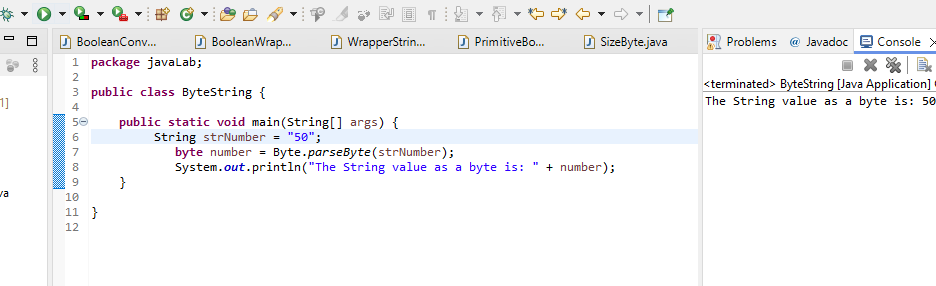
**c.** Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and Byte.MAX\_VALUE).



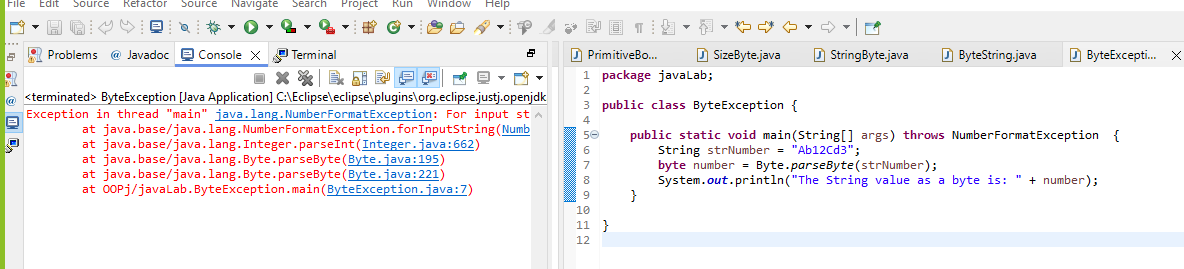
**d.** Declare a method-local variable number of type byte with some value and convert it to a String using the toString method. (Hint: Use Byte.toString(byte)).



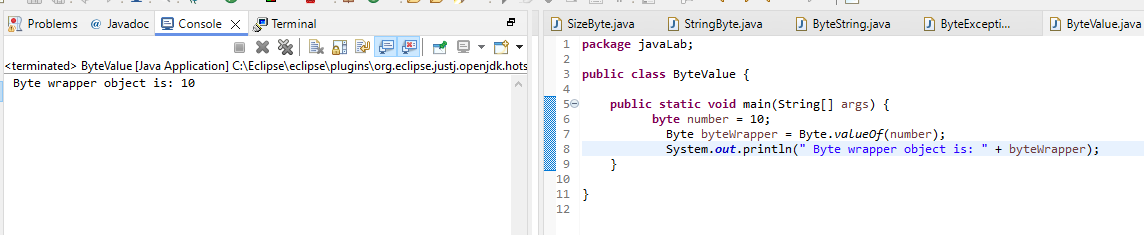
**e.** Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method. (Hint: Use Byte.parseByte(String)).



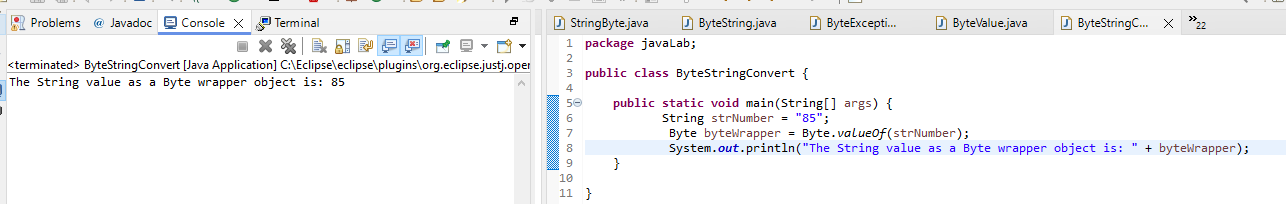
**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value. (Hint: parseByte method will throw a NumberFormatException).



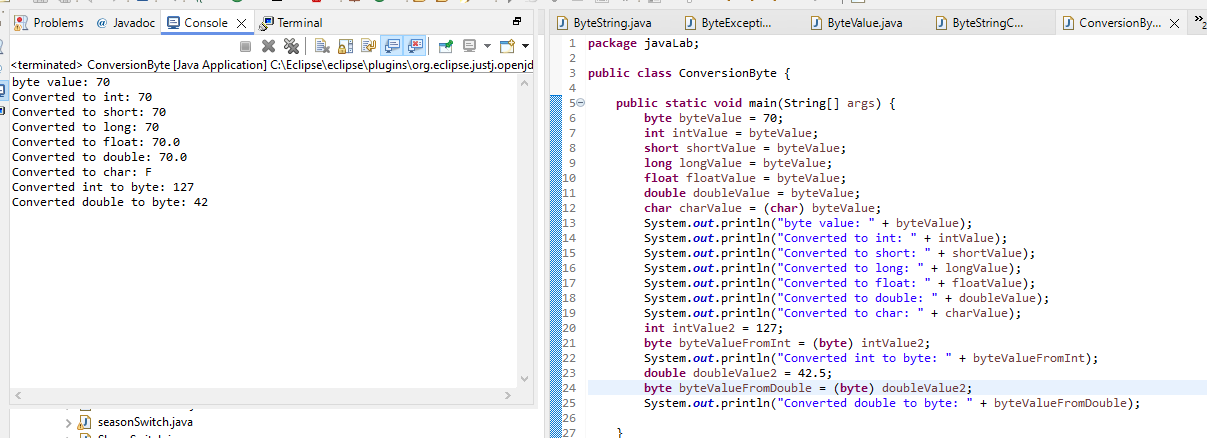
**g.** Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(byte)).



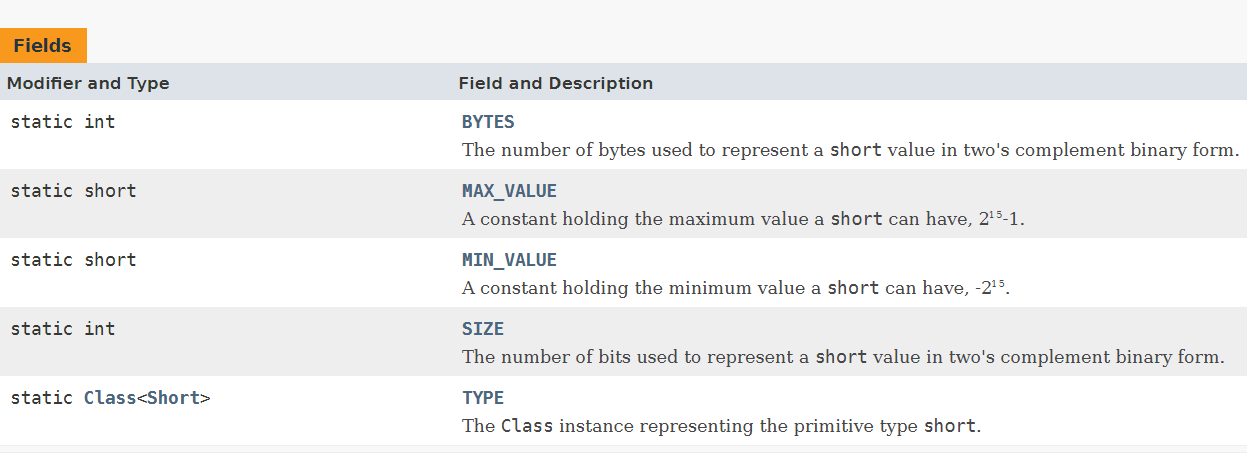
**h.** Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(String)).



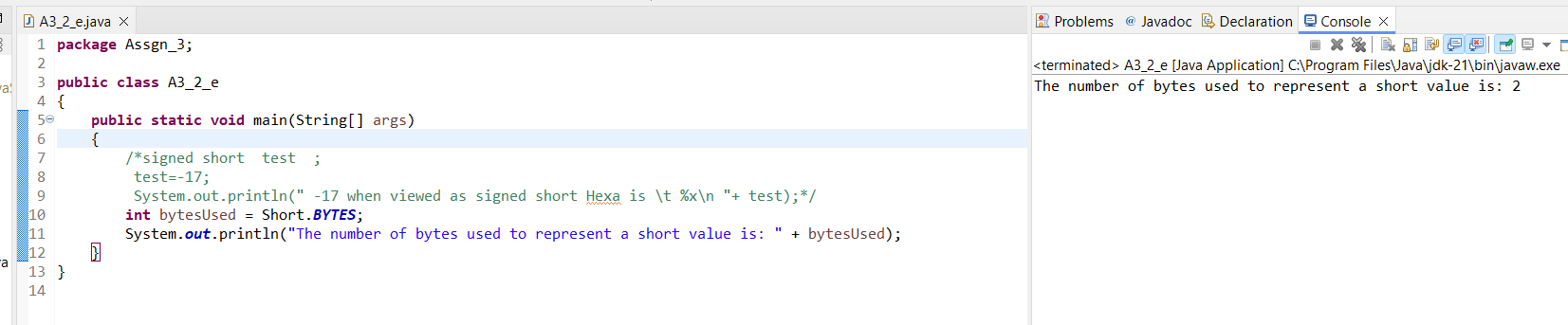
**i.** Experiment with converting a byte value into other primitive types or vice versa and observe the results.



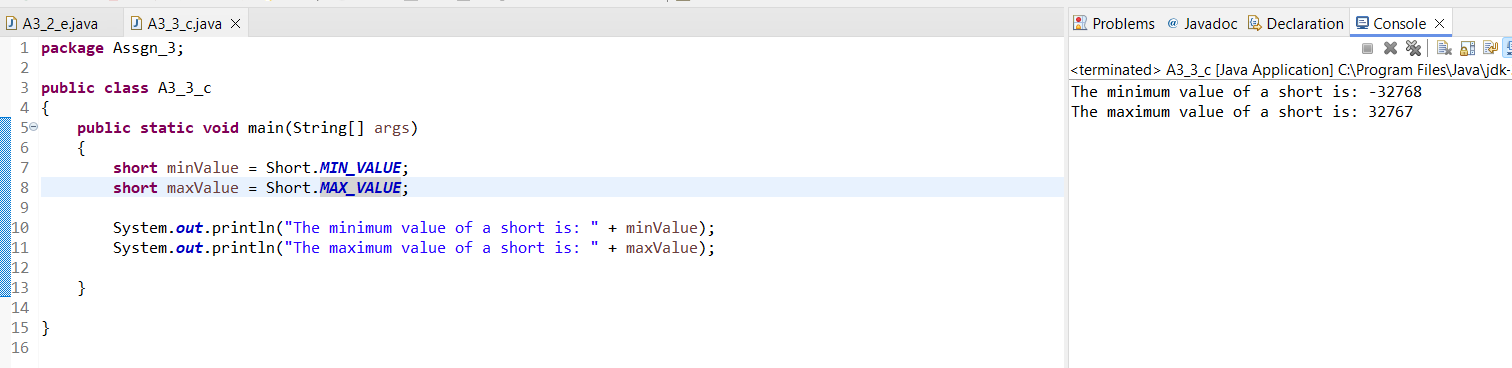
3. Working with **java.lang.Short**

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

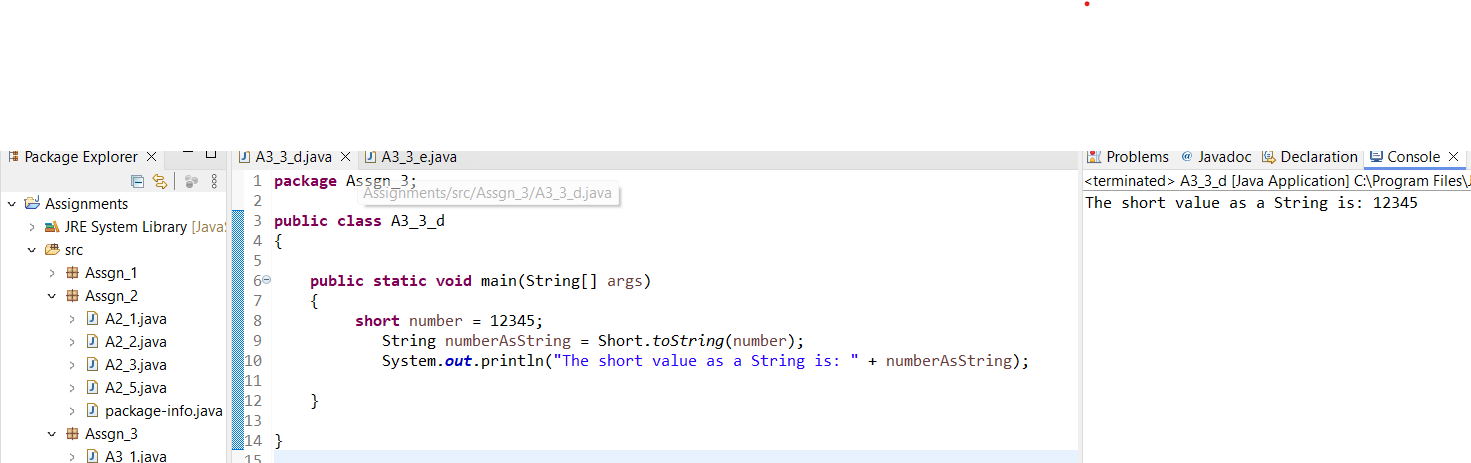
**b.** Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).



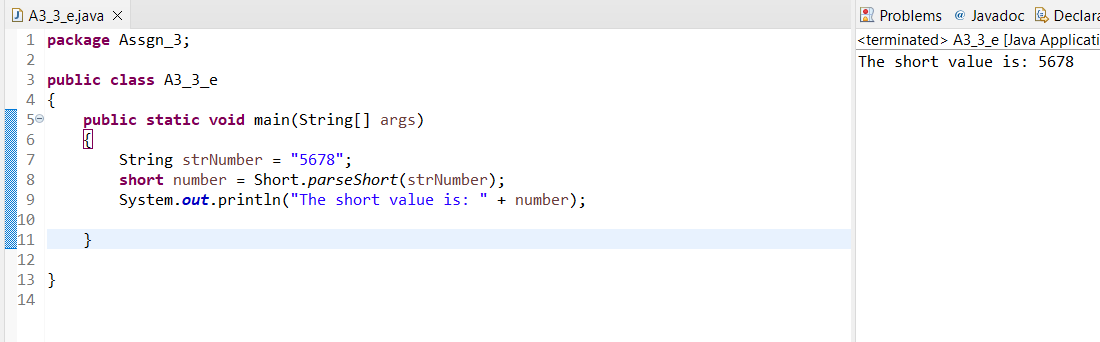
**c.** Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).



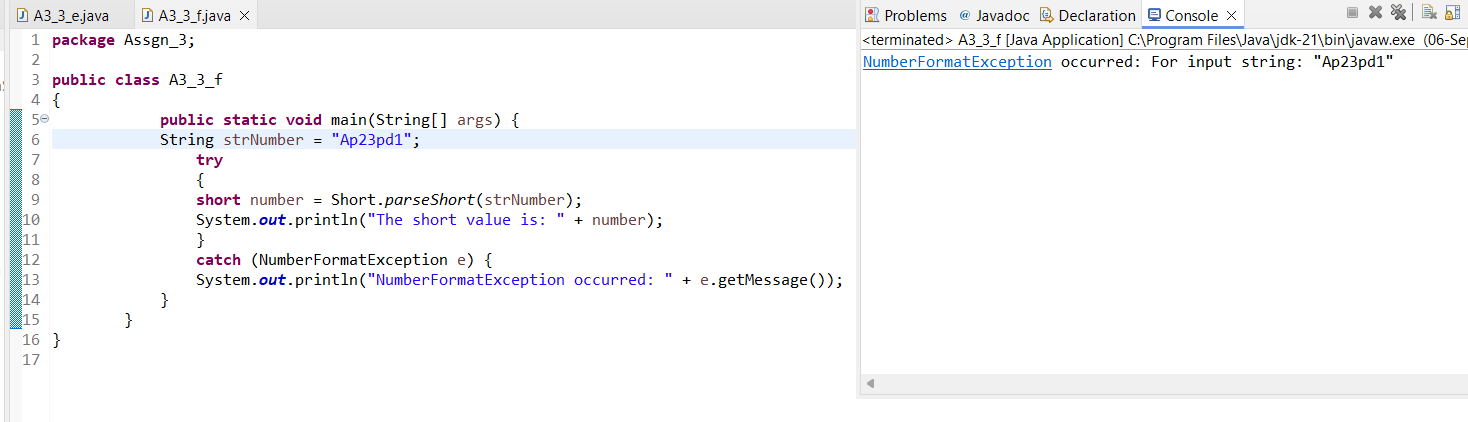
**d.** Declare a method-local variable number of type short with some value and convert it to a String using the toString method. (Hint: Use Short.toString(short)).

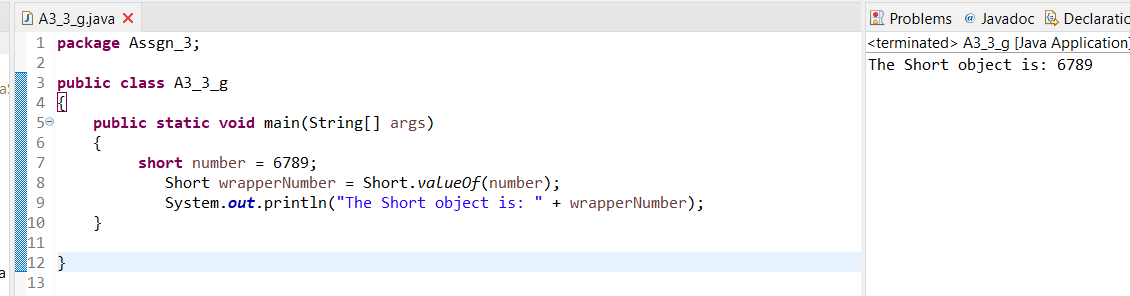


**e.** Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method. (Hint: Use Short.parseShort(String)).

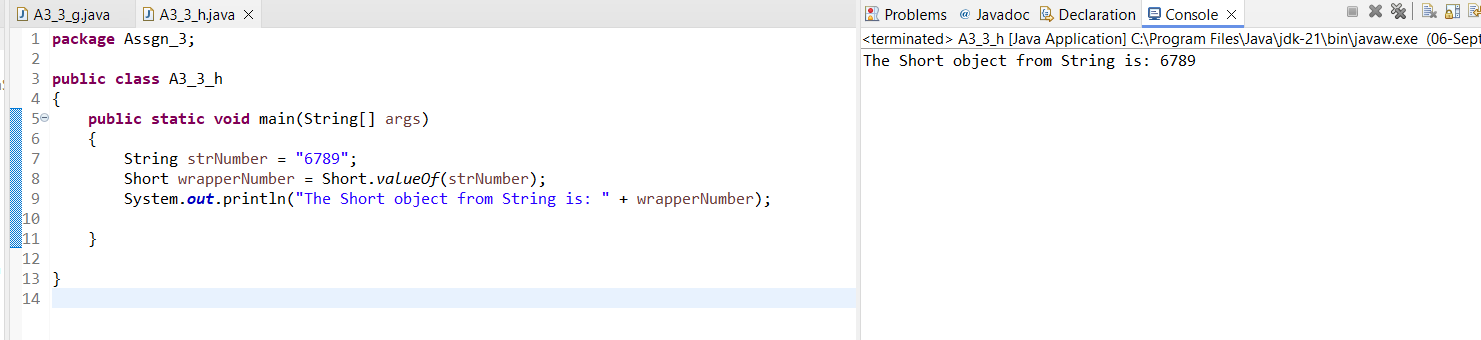


**f.** Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value. (Hint: parseShort method will throw a NumberFormatException).

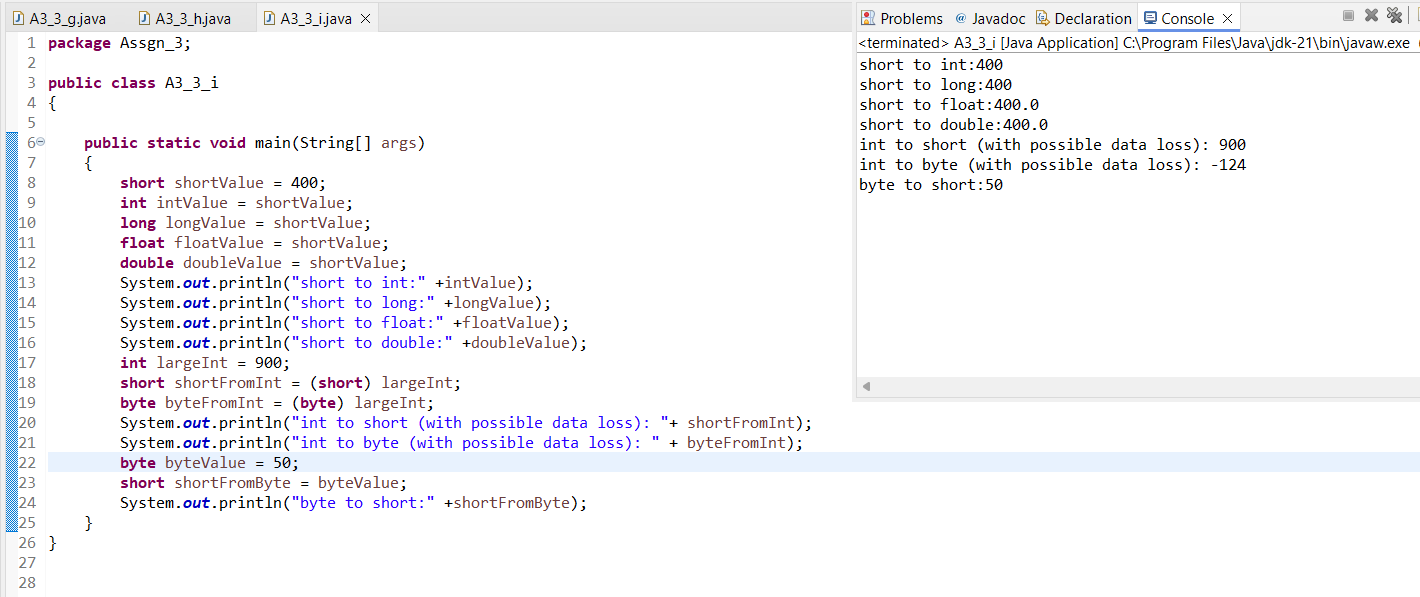
**g.** Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(short)).



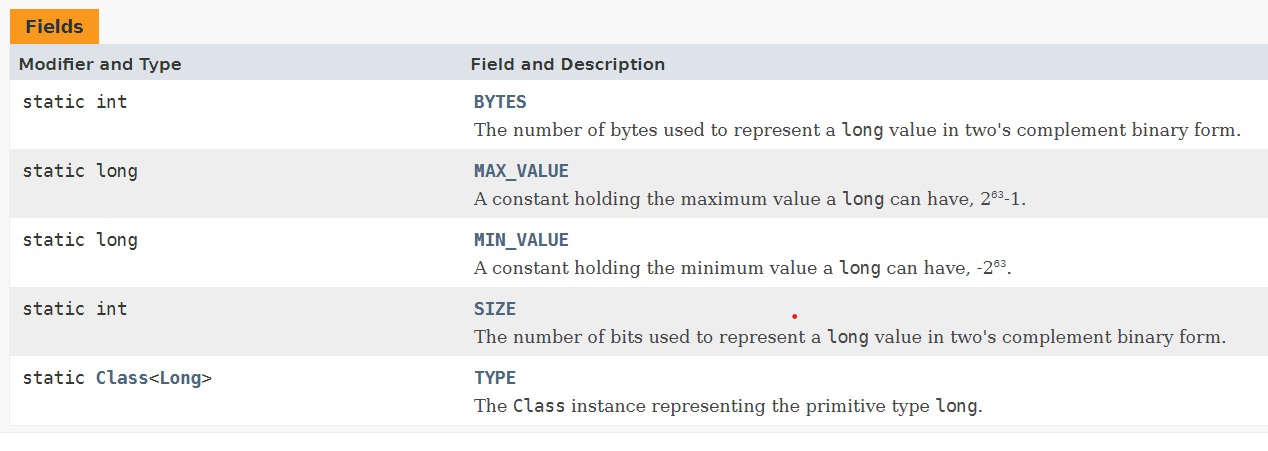
**h.** Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(String)).



**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).

**package** Assgn\_3;

**public** **class** A3\_4\_b {

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

**int** bytesUsed = Integer.***BYTES***;

System.***out***.println("Number of bytes used to represent an int: " + bytesUsed);

}

}

Output: Number of bytes used to represent an int: 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).

**package** Assgn\_3;

**public** **class** A3\_4\_c {

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

**int** minValue = Integer.***MIN\_VALUE***;

**int** maxValue = Integer.***MAX\_VALUE***;

System.***out***.println("Minimum value of an int: " + minValue);

System.***out***.println("Maximum value of an int: " + maxValue);

}

}

Output: Minimum value of an int: -2147483648

Maximum value of an int: 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)).

package Assgn\_3;

public class A3\_4\_d {

public static void main(String[] arg)

{

System.*out*.println("Integer.MAX\_VALUE = "

+ Integer.*MAX\_VALUE*);

}

}

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_e {

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

**int** a=1234;

**int** b=-1234;

String str1 = Integer.*toString*(a);

String str2 = Integer.*toString*(b);

System.***out***.println("String str1 = " + str1);

System.***out***.println("String str2 = " + str2);

}

}

Output: String str1 = 1234

String str2 = -1234

**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).

**package** Assgn\_3;

**public** **class** A3\_4\_f {

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

String numberStr = "12345";

**int** number = Integer.*parseInt*(numberStr);

System.***out***.println("The converted number is: " + number);

}

}

Output: The converted number is: 12345

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_g {

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

String strNumber = "Ap27Pd16";

**try** {

**int** number = Integer.*parseInt*(strNumber);

System.***out***.println("Converted number: " + number);

} **catch** (NumberFormatException e) {

System.***out***.println("Error: Invalid number format.");

}

}

}

Output: Error: Invalid number format.

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_h {

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

**int** number = 95;

Integer integerObject = Integer.*valueOf*(number);

System.***out***.println("Integer object: " + integerObject);

}

}

Output: Integer object: 95

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_i1 {

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

**int** num1 = 50;

**int** num2 = 20;

**int** sum = Integer.*sum*(num1, num2);

System.***out***.println("The sum of " + num1 + " and " + num2 + " is " + sum);

}

}

Output: The sum of 50 and 20 is 70

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_j {

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

**int** num1 = 50;

**int** num2 = 90;

**int** minValue = Integer.*min*(num1, num2);

**int** maxValue = Integer.*max*(num1, num2);

System.***out***.println("The minimum value between " + num1 + " and " + num2 + " is " + minValue);

System.***out***.println("The maximum value between " + num1 + " and " + num2 + " is " + maxValue);

}

}

Output: The minimum value between 50 and 90 is 50

The maximum value between 50 and 90 is 90

**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)).

**package** Assgn\_3;

**public** **class** A3\_4\_k {

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

**int** number = 9;

String binaryString = Integer.*toBinaryString*(number);

String octalString = Integer.*toOctalString*(number);

String hexString = Integer.*toHexString*(number);

System.***out***.println("Binary representation of " +number + " is " + binaryString);

System.***out***.println("Octal representation of " +number + " is " + octalString);

System.***out***.println("Hexadecimal representation of " +number+" is "+ hexString);

}

}

Output: Binary representation of 9 is 1001

Octal representation of 9 is 11

Hexadecimal representation of 9 is 9

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

**package** Assgn\_3;

**public** **class** A3\_4\_l {

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

**int** intValue=300;

**byte** byteValue = (**byte**) intValue;

System.***out***.println("Byte value: " + byteValue);

**short** shortValue = (**short**) intValue;

System.***out***.println("Short value: " + shortValue);

**long** longValue = (**long**) intValue;

System.***out***.println("Long value: " + longValue);

**float** floatValue = (**float**) intValue;

System.***out***.println("Float value: " + floatValue);

**double** doubleValue = (**double**) intValue;

System.***out***.println("Double value: " + doubleValue);

}

}

Output: Byte value: 44

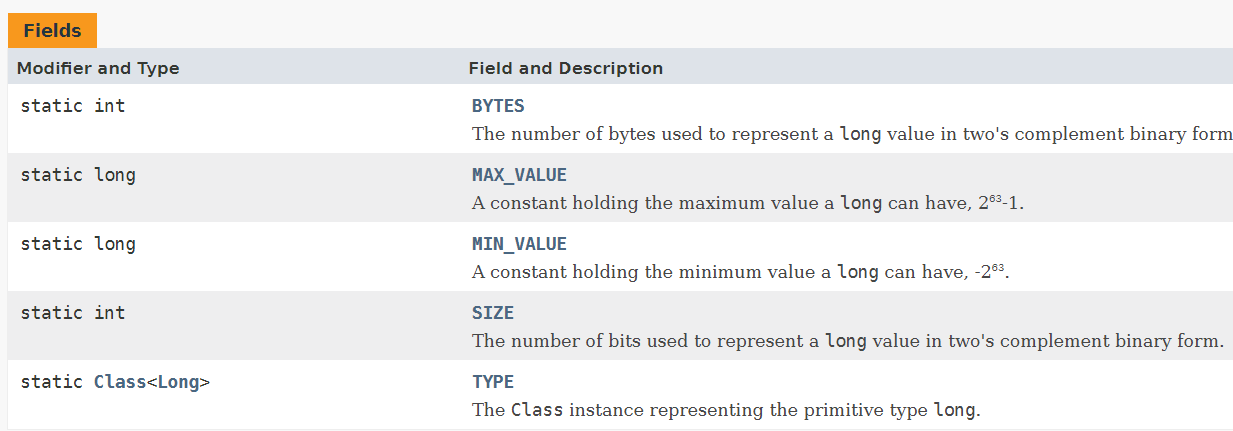
Short value: 300

Long value: 300

Float value: 300.0

Double value: 300.0

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).

**package** Assgn\_3;

**public** **class** A3\_5\_b {

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

**int** bytesUsed = Long.***BYTES***;

System.***out***.println("The number of bytes used to represent a long value: " + bytesUsed);

}

}

Output: The number of bytes used to represent a long value: 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).

**package** Assgn\_3;

**public** **class** A3\_5\_c {

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

**long** minValue = Long.***MIN\_VALUE***;

**long** maxValue = Long.***MAX\_VALUE***;

System.***out***.println("The minimum value of a long is: " + minValue);

System.***out***.println("The maximum value of a long is: " + maxValue);

}

}

Output: The minimum value of a long is: -9223372036854775808

The maximum value of a long is: 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)).

**package** Assgn\_3;

**public** **class** A3\_5\_d {

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

**long** number = 123456789L;

String numAsString = Long.*toString*(number);

System.***out***.println("The long value as a String is: " + numAsString);

}

}

Output: The long value as a String is: 123456789

**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)).

**package** Assgn\_3;

**public** **class** A3\_5\_e {

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

String strNumber = "2334445960";

**long** number = Long.*parseLong*(strNumber);

System.***out***.println("The String value as a long is: " + number);

}

}

Output: The String value as a long is: 2334445960

**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).

**package** Assgn\_3;

**public** **class** A3\_5\_f {

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

String strNumber = "Ab12Cd3";

**long** number = Long.*parseLong*(strNumber);

System.***out***.println("The String value as a long is: " + number);

}

}

Output: 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 Assgn\_3.A3\_5\_f.main(A3\_5\_f.java:5)

**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** Assgn\_3;

**public** **class** A3\_5\_g {

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

**long** number = 239375053L;

Long numberWrapper = Long.*valueOf*(number);

System.***out***.println("The Long object is: " + numberWrapper);

}

}

Output: The Long object is: 239375053

**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** Assgn\_3;

**public** **class** A3\_5\_h {

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

String strNumber = "1234567890";

Long longValue = Long.*valueOf*(strNumber);

System.***out***.println("The Long value is: " + longValue);

}

}

Output: The Long value is: 1234567890

**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)).

**package** Assgn\_3;

**public** **class** A3\_5\_i {

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

**long** num1 = 2020;

**long** num2 = 2025;

**long** sum = Long.*sum*(num1, num2);

System.***out***.println("The sum of " + num1 + " and " + num2 + " is: " + sum);

}

}

Output : The sum of 2020 and 2025 is: 4045

**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)).

**package** Assgn\_3;

**public** **class** A3\_5\_j {

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

**long** num1 = 1122;

**long** num2 = 5566;

**long** minValue = Long.*min*(num1, num2);

**long** maxValue = Long.*max*(num1, num2);

System.***out***.println("The minimum value between " + num1 + " and " + num2 + " is: " + minValue);

System.***out***.println("The maximum value between " + num1 + " and " + num2 + " is: " + maxValue);

}

}

Output: The minimum value between 1122 and 5566 is: 1122

The maximum value between 1122 and 5566 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)).

**package** Assgn\_3;

**public** **class** A3\_5\_k {

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

**long** number = 7;

String binaryString = Long.*toBinaryString*(number);

String octalString = Long.*toOctalString*(number);

String hexString = Long.*toHexString*(number);

System.***out***.println("Binary representation of " + number + " is: " + binaryString);

System.***out***.println("Octal representation of " + number + " is: " + octalString);

System.***out***.println("Hexadecimal representation of " + number + " is: " + hexString);

}

}

Output: Binary representation of 7 is: 111

Octal representation of 7 is: 7

Hexadecimal representation of 7 is: 7

**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.

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.

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()).

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).

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).