**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)).

**public class Demo {**

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

**boolean status = true;**

**String str = Boolean.toString(status);**

**System.out.println(str);**

**}**

**}**

**Output: true**

**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)).

**public class Boolstr {**

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

**String strStatus = "true";**

**boolean str = Boolean.parseBoolean(strStatus);**

**System.out.println(str);**

**}**

**}**

**Output: 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").

**public class Boolstr1 {**

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

**String strStatus = "1";**

**boolean str = "1".equals(strStatus);**

**System.*out*.println(str);**

**}**

**}**

**Output: true**

**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)).

**public class Boolstr2 {**

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

**boolean status = true;**

**Boolean statusStr = Boolean.*valueOf*(status);**

**System.*out*.println(statusStr);**

**}**

**}**

**Output: true**

**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)).

**public class Boolstr3 {**

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

**String status = "true";**

**Boolean strStatus = Boolean.*valueOf*(status);**

**System.*out*.println(strStatus);**

**}**

**}**

**Output: true**

**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).

**public class Byte1 {**

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

**System.*out*.println("The number of bytes used to represent a byte: " + Byte.*BYTES*);**

**}**

**}**

**Output: The number of bytes used to represent a byte: 1**

**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).

**public class Byte2 {**

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

**System.*out*.println( "Max Value of Byte: "+ Byte.*MAX\_VALUE* );**

**System.*out*.println( "Min Value of Byte:" + Byte.*MIN\_VALUE* );**

**}**

**}**

**Output: Max Value of Byte: 127**

**Min Value of Byte:-128**

**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)).

**public class Byte4 {**

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

**Byte number = 1;**

**String str = Byte.*toString*(number);**

**System.*out*.println(str);**

**}**

**}**

**Output: 1**

**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)).

**public class Byte3 {**

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

**String strNumber = "1";**

**byte num = Byte.*parseByte*(strNumber);**

**System.*out*.println(num);**

**}**

**}**

**Output: 1**

**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).

**public class Byte5 {**

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

**String strNumber = "Ab12c3";**

**Byte bt = Byte.*parseByte*(strNumber);**

**System.*out*.print(bt);**

**}**

**}**

**Output: Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12c3"**

**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.Byte.parseByte(Byte.java:195)**

**at java.base/java.lang.Byte.parseByte(Byte.java:221)**

**at com.example.Byte5.main(Byte5.java:7)**

**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)).

**public class Byte6 {**

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

**Byte number = 1;**

**Byte bt = Byte.valueOf(number);**

**System.out.println(bt);**

**}**

**}**

**Output: 1**

**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)).

**public class Byte8 {**

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

**String StrNumber = "1";**

**Byte bt = Byte.*valueOf*(StrNumber);**

**System.*out*.print(bt);**

**}**

**}**

**Output: 1**

**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).

**public class Short1 {**

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

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

**}**

**}**

**Output:** **Short Bytes value: 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).

**public class Short2 {**

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

**long a , b;**

**a = Short.*MIN\_VALUE*;**

**b=Short.*MAX\_VALUE*;**

**System.*out*.println(a);**

**System.*out*.println(b);**

**}**

**}**

**Output:**

**-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)).

**public class Short3 {**

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

**short obj = 1;**

**String strByte = Short.*toString*(obj);**

**System.*out*.print(strByte);**

**}**

**}**

**Output: 1**

**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)).

**public class Short4 {**

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

**String strNumber = "123";**

**Short num = Short.*parseShort*(strNumber);**

**System.*out*.println(num);**

**}**

**}**

**Output: 123**

**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).

**public class Short5 {**

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

**String StrNumber = "Ab12Cd3";**

**Short num = Short.*parseShort*(StrNumber);**

**System.*out*.println(num);**

**}**

**}**

**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.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 com.assignment2.Short5.main(Short5.java:7)**

**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 class Short6 {**

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

**Short num = 2;**

**Short num1 = Short.*valueOf*(num);**

**System.*out*.println(num1);**

**}**

**}**

**Output: 2**

**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 class Short7 {**

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

**String strNumber = "123";**

**Short num1 = Short.*valueOf*(strNumber);**

**System.*out*.println(num1);**

**}**

**}**

**Output: 123**

**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).

**public class Short8 {**

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

**System.out.println("Bytes used to represent: " +Integer.BYTES);**

**}**

**}**

**Output: Bytes used to represent: 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).

**public class Integer2 {**

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

**long a , b;**

**a = Integer.*MIN\_VALUE*;**

**b=Integer.*MAX\_VALUE*;**

**System.*out*.println(a);**

**System.*out*.println(b);**

**}**

**}**

**Output: -2147483648**

**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)).

**public class Integer3 {**

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

**int number = 123;**

**String strNum = Integer.*toString*(number);**

**System.*out*.println(strNum);**

**}**

**}**

**Output: 123**

**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)).

**public class Integer4 {**

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

**String StrNumber= 123;**

**int num1 = Integer.*parseInt*(StrNumber);**

**System.*out*.println(num1);**

**}**

**}**

**Output: 123**

**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).

**public class Integer5 {**

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

**int number = “Ab12Cd3”;**

**int num1 = Integer.*parseInt*(StrNumber);**

**System.*out*.println(num1);**

**}**

**}**

**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.Integer.parseInt(Integer.java:662)**

**at java.base/java.lang.Integer.parseInt(Integer.java:778)**

**at com.assignment2.Integer4.main(Integer4.java:7)**

**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)).

**public class Integer6 {**

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

**int number = 123;**

**int num1 = Integer.*valueOf*(number);**

**System.*out*.println(num1);**

**}**

**}**

**Output: 123**

**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)).

**public class Integer7 {**

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

**String strNumber = "123";**

**int num1 = Integer.*valueOf*(strNumber);**

**System.*out*.println(num1);**

**}**

**}**

**Output: 123**

**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)).

**public class Integer8 {**

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

**int a= 10;**

**int b = 20;**

**int result = Integer. *sum*( a , b );**

**System.*out*.println("Sum of two numbers 10 and 20 is:" +result );**

**}**

**}**

**Output: Sum of two numbers 10 and 20 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)).

**public class Integer9 {**

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

**int a= 10;**

**int b = 20;**

**int mini = Integer.*min*( a , b );**

**int maxi = Integer.*max*( a , b );**

**System.*out*.println("Minimum and maximum values are:" +mini +" & "+maxi );**

**}**

**}**

**Output: Minimum and maximum values are:10 & 20**

**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)).

**public class Integer10 {**

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

**int a = 7;**

**String bin = Integer.*toBinaryString*( a );**

**String oct = Integer.*toOctalString*( a );**

**String hex = Integer.*toHexString*( a );**

**System.*out*.println("Binary value of 7"+bin );**

**System.*out*.println("Octal value of 7"+oct );**

**System.*out*.println("Hexadecimal value of 7"+hex );**

**}**

**}**

**Output**

**Binary value of 7111**

**Octal value of 77**

**Hexadecimal value of 77**

**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).

**public class Long1 {**

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

**System.*out*.println("Bytes used to represent:" +Long.*BYTES*);**

**}**

**}**

**Output: Bytes used to represent: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).

**public class Long2 {**

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

**long a , b;**

**a = Long.*MIN\_VALUE*;**

**b=Long.*MAX\_VALUE*;**

**System.*out*.println(a);**

**System.*out*.println(b);**

**}**

**}**

**Output**

**-9223372036854775808**

**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)).

**public class Long3 {**

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

**long number = 123;**

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

**System.*out*.println(strNum);**

**}**

**}**

**Output: 123**

**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 class Long4 {**

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

**String number = "123";**

**Long strNum = Long.*parseLong*(number);**

**System.*out*.println(strNum);**

**}**

**}**

**Output: 123**

**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 class Long5 {**

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

**String strNumber = "Ab12Cd3";**

**Long strNum = Long.*parseLong*(strNumber);**

**System.*out*.println(strNum);**

**}**

**}**

**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 com.assignment2.Long3.main(Long3.java:7)**

**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)).

**public class Long6 {**

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

**String number = "123";**

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

**System.*out*.println(strNum);**

**}**

**}**

**Output: 123**

**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)).

**public class Long7 {**

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

**String strNumber = "123456789";**

**long strNum = Long.*valueOf*(strNumber);**

**System.*out*.println(strNum);**

**}**

**}**

**Output: 123456789**

**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 Long8 {**

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

**long a = 1123;**

**long b = 5566;**

**long result = Long.*sum*(a, b);**

**System.*out*.println("Long variables a and b sum is "+result);**

**}**

**}**

**Output: Long variables a and b sum is 6689**

**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)).

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

**long a = 1122;**

**long b = 5566;**

**long mini = Long.*min*(a,b);**

**long maxi = Long.*max*(a,b);**

**System.*out*.println("Minimum and Maximum numbers of a and b minimum: "+mini+" maximum: "+maxi );**

**}**

**Output: Minimum and Maximum numbers of a and b minimum: 1122 maximum: 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)).

**public class Long10 {**

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

**long a = 7;**

**String bin = Long.*toBinaryString*( a );**

**String oct = Long.*toOctalString*( a );**

**String hex = Long.*toHexString*( a );**

**System.*out*.println("Binary value of 7"+bin );**

**System.*out*.println("Octal value of 7"+oct );**

**System.*out*.println("Hexadecimal value of 7"+hex );**

**}**

**}**

**Output:**

**Binary value of 7111**

**Octal value of 77**

**Hexadecimal value of 77**

**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).

**public class Float1 {**

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

**System.*out*.println("Bytes used to represent:" +Float.*BYTES*);**

**}**

**}**

**Output: Bytes used to represent: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 class Float2 {**

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

**float a , b;**

**a = Float.*MIN\_VALUE*;**

**b=Float.*MAX\_VALUE*;**

**System.*out*.println(a);**

**System.*out*.println(b);**

**}**

**}**

**Output: 1.4E-45**

**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)).

**public class Float3 {**

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

**float number = 12.345f;**

**String num = Float.*toString*(number);**

**System.*out*.println(num);**

**}**

**}**

**Output: 12.345**

**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)).

**public class Float3 {**

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

**String strNumber = "1234.44f";**

**float num = Float.*parseFloat*(strNumber);**

**System.*out*.println(num);**

**}**

**}**

**Output: 1234.44**

**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).

**public class Float4 {**

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

**String strNumber = "** **Ab12Cd3";**

**float num = Float.*parseFloat*(strNumber);**

**System.*out*.println(num);**

**}**

**}**

**Output: Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"**

**at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)**

**at java.base/jdk.internal.math.FloatingDecimal.parseFloat(FloatingDecimal.java:122)**

**at java.base/java.lang.Float.parseFloat(Float.java:556)**

**at com.assignment2.Float5.main(Float5.java:7)**

**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)).

**public class Float6 {**

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

**float number = 1234.5f;**

**float flonum = Float.*valueOf*(number);**

**System.*out*.println(flonum);**

**}**

**}**

**Output: 1234.5**

**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)).

**public class Float7 {**

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

**String strNumber = "1234.5f";**

**float flonum = Float.*valueOf*(strNumber);**

**System.*out*.println(flonum);**

**}**

**}**

**Output: 1234.5**

**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)).

**public class Float8 {**

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

**float a= 112.3f;**

**float b = 984.5f;**

**float result = Float.*sum*(a, b);**

**System.*out*.println("Sum of a and b of float variables: "+result );**

**}**

**}**

**Output: Sum of a and b of float variables: 1096.8**

**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)).

**public class Float10 {**

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

**float a= 112.2f;**

**float b = 556.6f;**

**float mini = Float.*min*( a , b );**

**float maxi = Float.*max*( a , b );**

**System.*out*.println("Minimum and maximum values are:" +mini +" & "+maxi );**

**}**

**}**

**Output: Minimum and maximum values are:112.2 & 556.6**

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

public class Float10 {

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

**float val = -25.0f;**

**double sqrt = Math.*sqrt*(val);**

**System.*out*.println("Square root value of: "+sqrt);**

**}**

**}**

**Output: Square root value of: NaN**

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

**public class Float11 {**

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

**float a = 0.0f;**

**float b = 0.0f;**

**float result = a/b;**

**System.*out*.println("Float a and b division: "+result);**

**}**

**}**

**Output: Float a and b division: NaN**

**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).

**public class Double1 {**

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

**System.*out*.println("Bytes used to represent:" +Double.*BYTES*);**

**}**

**}**

**Output: Bytes used to represent:8**

**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).

**public class Double2 {**

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

**double a , b;**

**a = Double.*MIN\_VALUE*;**

**b=Double.*MAX\_VALUE*;**

**System.*out*.println(a);**

**System.*out*.println(b);**

**}**

**}**

**Output:**

**4.9E-324**

**1.7976931348623157E308**

**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)).

**public class Double3 {**

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

**double number = 3.1415926535;**

**String douNum = Double.*toString*(number);**

**System.*out*.println(douNum);**

**}**

**}**

**Output: 3.1415926535**

**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)).

**public class Double4 {**

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

**String strNumber = "3.1415926535";**

**double douNum = Double.*parseDouble*(strNumber);**

**System.*out*.println(douNum);**

**}**

**}**

**Output: 3.1415926535**

**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).

**public class Double5 {**

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

**String strNumber = "3.1415926535";**

**double douNum = Double.*parseDouble*(strNumber);**

**System.*out*.println(douNum);**

**}**

**}**

**Output:**

**Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"**

**at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)**

**at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)**

**at java.base/java.lang.Double.parseDouble(Double.java:792)**

**at com.assignment2.Double4.main(Double4.java:8)**

**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)).

**public class Double6 {**

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

**double number = "3.1415926535";**

**double douNum = Double.*valueOf*(number);**

**System.*out*.println(douNum);**

**}**

**}**

**Output: 3.1415926535**

**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)).

**public class Double7 {**

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

**String strNumber = "3.1415926535";**

**double douNum = Double.*valueOf*(strNumber);**

**System.*out*.println(douNum);**

**}**

**}**

**Output: 3.1415926535**

**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)).

**public class Double8 {**

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

**double a= 112.3f;**

**double b = 984.5f;**

**double result = Double.*sum*(a, b);**

**System.*out*.println("Sum of a and b of float variables: "+result );**

**}**

**}**

**Output: Sum of a and b of float variables: 1096.8000030517578**

**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)).

**public class Double9 {**

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

**double a= 112.2f;**

**float b = 556.6f;**

**double mini = Double.*min*( a , b );**

**double maxi = Double.*max*( a , b );**

**System.*out*.println("Minimum and maximum values are:" +mini +" & "+maxi );**

**}**

**}**

**Output:**

**Minimum and maximum values are:112.19999694824219 & 556.5999755859375**

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

**public class Double10 {**

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

**double val = -25.0f;**

**double sqrt = Math.*sqrt*(val);**

**System.*out*.println("Square root value of: "+sqrt);**

**}**

**}**

**Output: Square root value of: NaN**

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

**public class Double11 {**

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

**double a = 0.0f;**

**double b = 0.0f;**

**double result = a/b;**

**System.*out*.println("Float a and b division: "+result);**

**}**

**}**

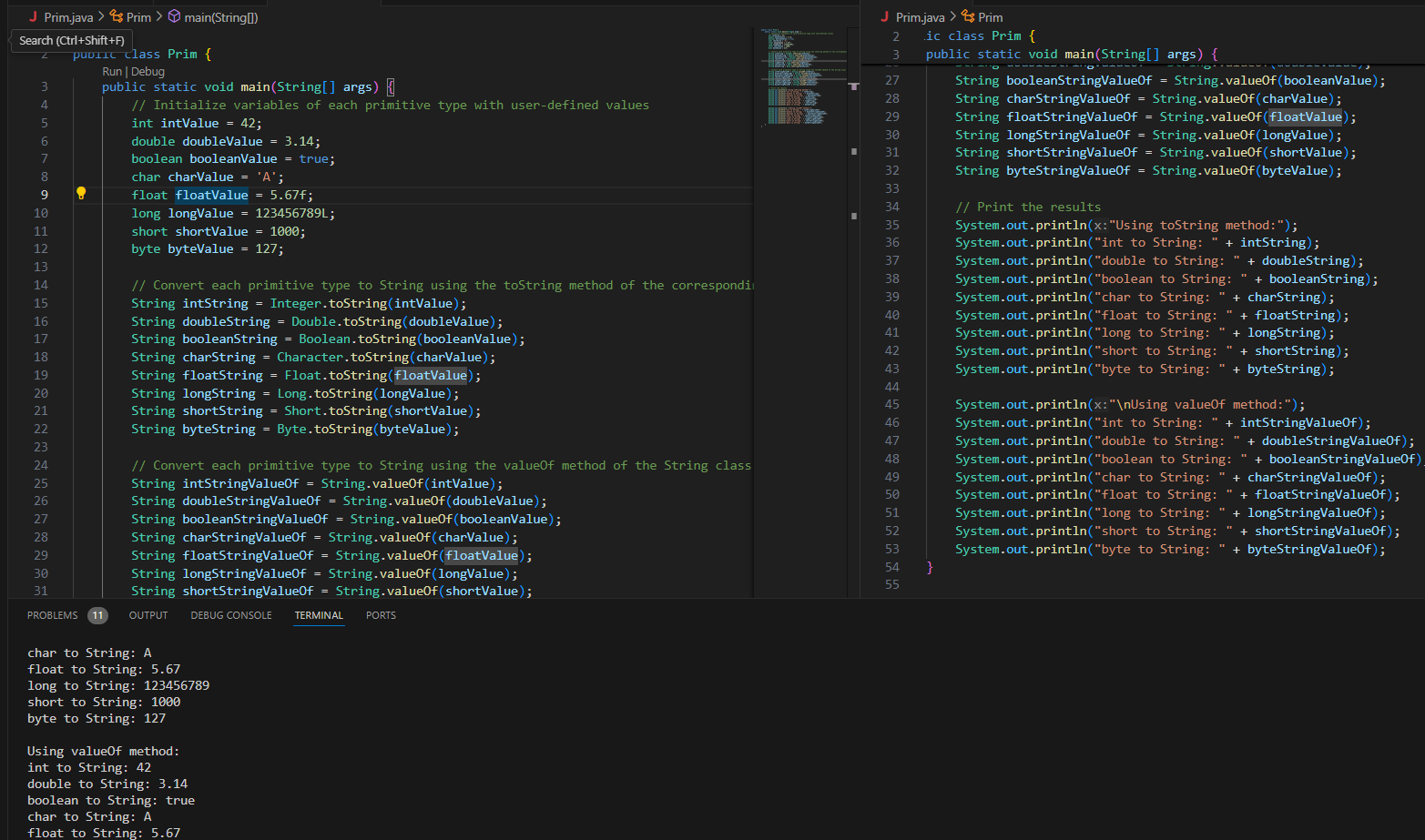
**Output: Float a and b division: NaN**

**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****

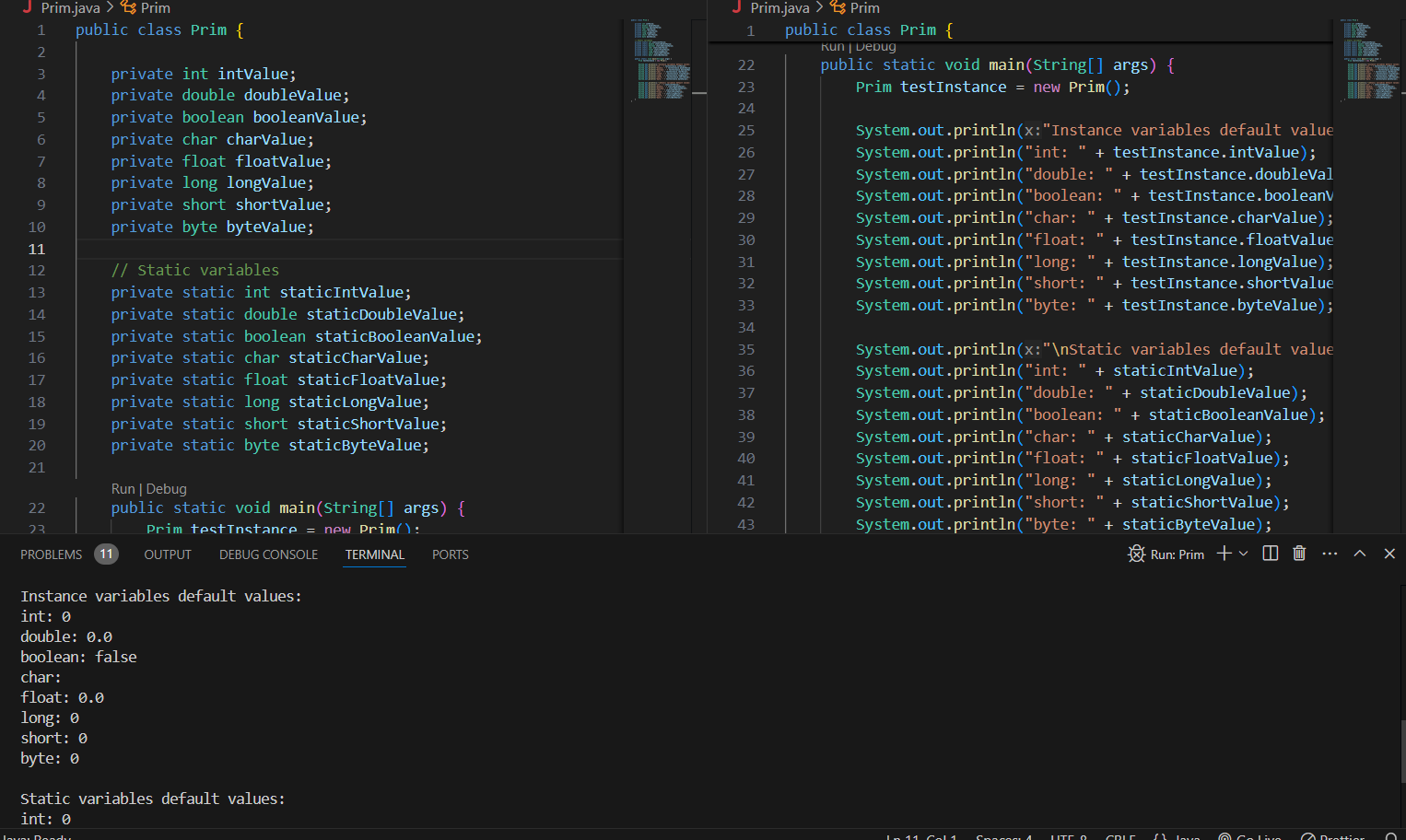
Initialize a variable of each primitive type with a user-defined value and convert it into String:

* + 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).

