

Q5] Working with java.lang.Long

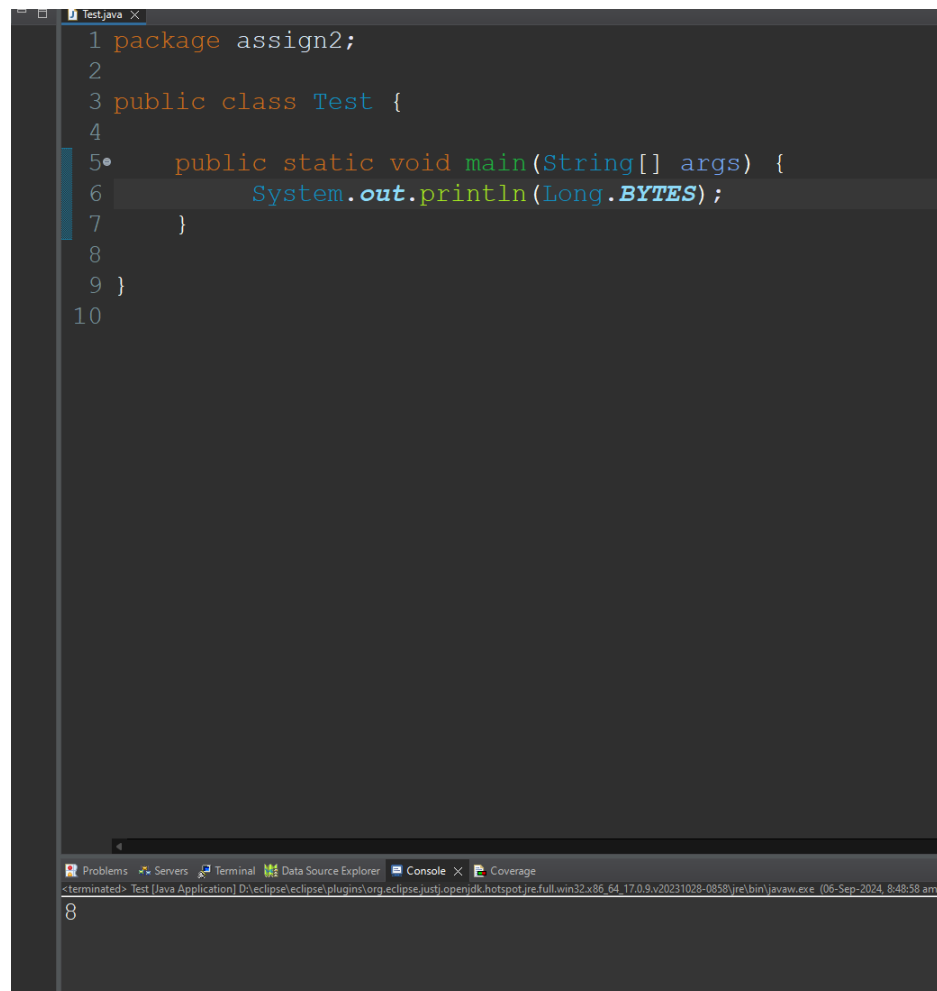
- a) Explore the Java API documentation for `java.lang.Long` and observe its modifiers and super types.

⇒ static long → MAX_VALUE

static long → MIN_VALUE

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

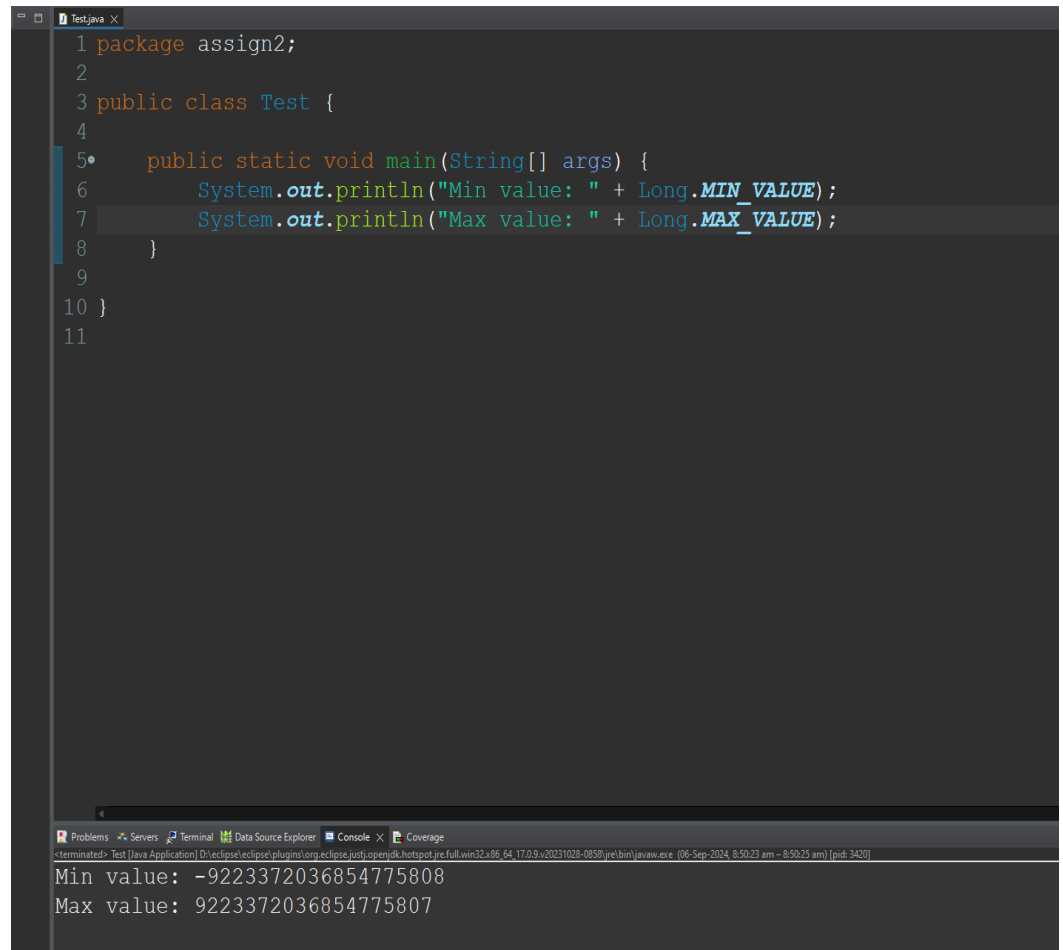
⇒



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         System.out.println(Long.BYTES);
7     }
8
9 }
10
```

The screenshot shows the Eclipse IDE interface. The main editor window displays a Java file named `Test.java` with the following code: `package assign2;`, `public class Test {`, `public static void main(String[] args) {`, `System.out.println(Long.BYTES);`, and `}`. The `main` method is highlighted with a blue selection bar. The bottom of the IDE shows the `Console` tab, which contains the output `8`, indicating that a long value is represented by 8 bytes. The status bar at the very bottom shows the file path and a timestamp: `D:\eclipse\workspace\assign2\src\main\java\assign2\Test.java (06-Sep-2024, 8:48:58 am)`.

- 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).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         System.out.println("Min value: " + Long.MIN_VALUE);
7         System.out.println("Max value: " + Long.MAX_VALUE);
8     }
9
10 }
11
```

Min value: -9223372036854775808
Max value: 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)).



```
Test.java X
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long number = 123456789L;
7         String numberString = Long.toString(number);
8         System.out.println(numberString);
9     }
10
11 }
12
```

Problems Servers Terminal Data Source Explorer Console Coverage
<terminated> Test [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\java.exe [06-Sep-2024, 8:52:01 am - 8:52:03 am] [pid: 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)).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         String strNumber = "123456789";
7         long number = Long.parseLong(strNumber);
8         System.out.println(number);
9     }
10
11 }
12
```

123456789

- 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).



The screenshot shows an IDE with a Java file named `Test.java` in the package `assign2`. The code defines a `Test` class with a `main` method. Inside `main`, a `String` variable `strNumber` is assigned the value `"Ab12Cd3"`, and a `long` variable `number` is assigned the result of `Long.parseLong(strNumber)`. The `System.out.println(number)` statement is also present. The IDE's console window at the bottom displays a `java.lang.NumberFormatException` error, indicating that the input string `"Ab12Cd3"` is not a valid long number.

```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         String strNumber = "Ab12Cd3";
7         long number = Long.parseLong(strNumber);
8         System.out.println(number);
9     }
10
11 }
12
```

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:711)
at java.base/java.lang.Long.parseLong(Long.java:836)
at assign2.Test.main(Test.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)`).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long number = 123456789L;
7         Long wrapperNumber = Long.valueOf(number);
8         System.out.println(wrapperNumber);
9     }
10
11 }
12
```

Problems Servers Terminal Data Source Explorer Console Coverage
<terminated> Test [Java Application] D:\eclipse\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.9.v20231028-0850\jre\bin\javaw.exe (06-Sep-2024, 9:12:00 am - 9:12:01 am) [pid: 16960]
123456789

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 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         String strNumber = "123456789";
7         Long wrapperNumber = Long.valueOf(strNumber);
8         System.out.println(wrapperNumber);
9     }
10
11 }
12
```

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)).



```
Test.java
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long num1 = 1123;
7         long num2 = 9845;
8         long sum = Long.sum(num1, num2);
9         System.out.println(sum);
10    }
11
12 }
13
```

Problems Servers Terminal Data Source Explorer Console Coverage

<terminated> Test [Java Application] D:\eclipse\workspace\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64-17.0.3.v20231020-0858\jre\bin\java.exe (08-Sep-2024, 9:18:43 am) [pid: 5832]

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)).



```
Test.java
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long num1 = 1122;
7         long num2 = 5566;
8         long min = Long.min(num1, num2);
9         long max = Long.max(num1, num2);
10        System.out.println("Min: " + min);
11        System.out.println("Max: " + max);
12    }
13
14 }
15
```

Problems Servers Terminal Data Source Explorer Console Coverage

<terminated> Test [Java Application] D:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (06-Sep-2024, 9:20:15 am - 9:20:17 am)

Min: 1122
Max: 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)`).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long number = 7;
7         String binaryString = Long.toBinaryString(number);
8         String octalString = Long.toOctalString(number);
9         String hexString = Long.toHexString(number);
10
11         System.out.println("Binary: " + binaryString);
12         System.out.println("Octal: " + octalString);
13         System.out.println("Hexadecimal: " + hexString);
14     }
15
16 }
17
```

Binary: 111
Octal: 7
Hexadecimal: 7

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



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         long longValue = 123456789L;
7         int intValue = (int) longValue;
8         System.out.println("Long value: " + longValue);
9         System.out.println("Converted to int: " + intValue);
10    }
11
12 }
13
```

Long value: 123456789
Converted to int: 123456789