

## Q6] Working with java.lang.Float

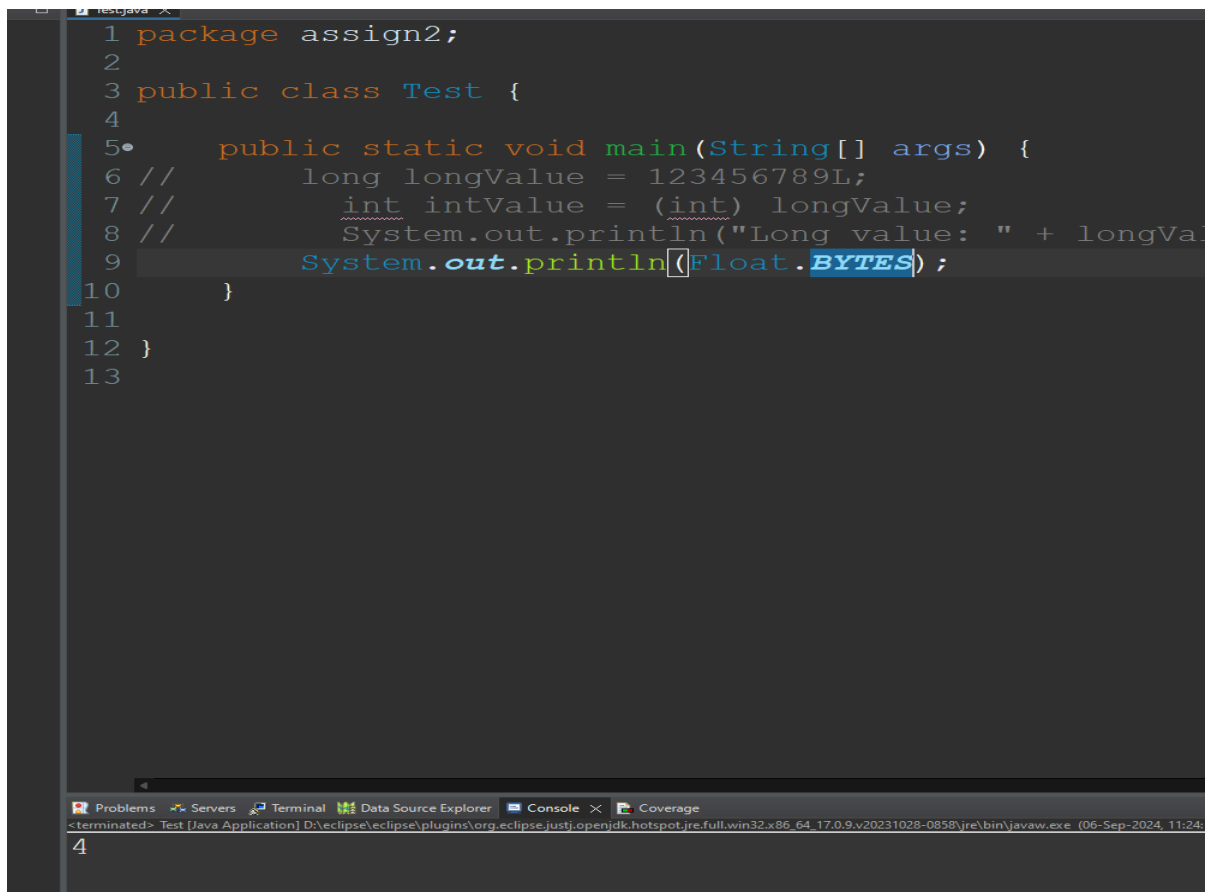
- a) a. Explore the Java API documentation for java.lang.Float and observe its modifiers and super types.

⇒ static float → MIN\_VALUE

static float → NaN

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

⇒



```
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(Float.BYTES);
10    }
11
12 }
13
```

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).



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

Min value: 1.4E-45  
Max value: 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)).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float number = 3.14f;
7         String numberString = Float.toString(number);
8         System.out.println(numberString);
9     }
10
11 }
12
```

3.14

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)).



```
Test.java x
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         String strNumber = "3.14";
7         float number = Float.parseFloat(strNumber);
8         System.out.println(number);
9     }
10
11 }
12
```

3.14

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`).



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

The screenshot shows an IDE window with a Java file named 'Test.java'. The code defines a class 'Test' with a 'main' method. Inside 'main', a string 'Ab12Cd3' is assigned to 'strNumber', and 'Float.parseFloat(strNumber)' is called to convert it to a float, which is then printed. The console output at the bottom shows a 'java.lang.NumberFormatException: For input string: "Ab12Cd3"' exception, with the stack trace pointing to the 'parseFloat' method in 'Float.java' and the 'main' method in 'Test.java'.

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)`).



```
Test.java x
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float number = 3.14f;
7         Float wrapperNumber = Float.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\workspace\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0850\jre\bin\javaw.exe (06-Sep-2024, 11:49:23 am - 11:49:24 a

3.14

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)).



```
Test.java
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         String strNumber = "3.14";
7         Float wrapperNumber = Float.valueOf(strNumber);
8         System.out.println(wrapperNumber);
9     }
10
11 }
12
```

3.14

- 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)).



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float num1 = 112.3f;
7         float num2 = 984.5f;
8         float sum = Float.sum(num1, num2);
9         System.out.println(sum);
10    }
11
12 }
13
```

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)`).





```
Testjava x
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float num1 = 112.2f;
7         float num2 = 556.6f;
8         float min = Float.min(num1, num2);
9         float max = Float.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\workspace\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\javaw.exe (06-Sep-2024, 11:53:17)

Min: 112.2  
Max: 556.6

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



```
Test.java x
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float value = -25.0f;
7         double squareRoot = Math.sqrt(value);
8         System.out.println("Square root: " + squareRoot);
9     }
10
11 }
12
```

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, 11:54:26 am - 11:54:28 am) [pid: 5]

Square root: NaN

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



```
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float num1 = 0.0f;
7         float num2 = 0.0f;
8         float result = num1 / num2;
9         System.out.println("Result of division: " + result);
10    }
11
12 }
13
```

Result of division: NaN

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



```
Test.java x
1 package assign2;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         float floatValue = 123.456f;
7         int intValue = (int) floatValue;
8         System.out.println("Float value: " + floatValue);
9         System.out.println("Converted to int: " + intValue);
10    }
11
12 }
13
```

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.v20231028-0858\jre\bin\javaw.exe (06-Sep-2024, 11:56:49 am - 11:56:50 am) [pid: 16128]

Float value: 123.456  
Converted to int: 123