

**Name: 117\_Swapnil Dhotre \_KH**

## **OOPJ Assignment - 2**

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

- a. Explore the Java API documentation for `java.lang.Boolean` 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.

```
J ques1b.java > ...
1  class ques1b{
2    Run | Debug
3    public static void main(String[] args){
4      boolean status =true;
5      String bool = Boolean.toString(status) ;
6      System.out.println(x:"status");
7    }

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1b
status
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

- c. Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method.

The screenshot shows a Java code editor with a dark theme. A terminal window is integrated at the bottom, showing the execution of a Java program. The code in the editor is:

```
J ques1c.java > ...
1  | class ques1c{
2  |     Run | Debug
3  |     public static void main(String[] args){
4  |         String strstatus = "true" ;
5  |         boolean strbool= Boolean.parseBoolean(strstatus);
6  |         System.out.println(strbool);
7  |     }
8
9
10
11
```

The terminal window displays the following output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1d
false
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

- d. Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean.

The screenshot shows a Java code editor window with a dark theme. The code in q1d.java is:

```
J q1d.java > ...
1 class q1d{
2     Run | Debug
3     public static void main(String[] args){
4         String strStatus= "0";
5         boolean strbool = Boolean.parseBoolean(strStatus);
6         System.out.println(strbool);
7     }
8 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q1d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q1d
false
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

- 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().

The screenshot shows a Java code editor with a dark theme. A code completion tooltip is displayed over the line of code: "boolean status = true ; Boolean.valueOf(status);". The tooltip contains the following options:

- Boolean.TRUE
- Boolean.FALSE
- Boolean.valueOf(status)
- Boolean.parseBoolean(status)

The code editor interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. Below the editor, a terminal window shows the execution of the Java program:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1e
true
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> $
```

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().

The screenshot shows a code editor interface with a dark theme. At the top, there's a status bar with the text "ques1f.java > ...". Below the status bar is the code area containing the following Java code:

```
J ques1f.java > ...
1  class ques1f{
2      Run | Debug
3      public static void main(String[] args){
4          String strStatus = "true";
5          Boolean.valueOf(strStatus);
6          System.out.println(strStatus);
7      }
8
9
10
```

Below the code area is a terminal window showing the execution of the Java program:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1f
true
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

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

The screenshot shows a Java code editor with a dark theme. A file named `ques1g.java` is open, containing the following code:

```
J ques1g.java > ...
1  class ques1g{
2    public static void main(String[] args){
3      int x=1;
4      Boolean bool=x!=0;
5      // res=x+y;
6      System.out.println(bool);
7    }
8
9
10 }
```

The code defines a class `ques1g` with a `main` method. Inside the `main` method, an integer `x` is set to 1, a boolean `bool` is assigned the value of `x != 0`, and the result is printed to the console. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, with TERMINAL selected. The terminal window shows the command-line interface with the following output:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac ques1g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java ques1g
true
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

## 2. Working with java.lang.Byte

- a. Explore the Java API documentation for java.lang.Byte 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.

The screenshot shows a Java code editor with a dark theme. A single file named 'q2b.java' is open, containing the following code:

```
J q2b.java > q2b > main(String[])
1 class q2b{
2     public static void main(String[] args){
3         Byte b=Byte.BYTES;
4         System.out.println(b);
5     }
6 }
```

Below the code editor, there is a terminal window showing the execution of the program:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac q2b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java q2b
1
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> 
```

c. Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields.

The screenshot shows a Java code editor with a terminal window below it. The code in the editor is:

```
q2c.java > ...
1  public class q2c {
2      |  public static void main(String[] args){
3          |      byte min= Byte.MIN_VALUE;
4          |      byte max= Byte.MAX_VALUE;
5          |      System.out.println("Min Value: "+min);
6          |      System.out.println("Max Value: "+max);
7      }
8  }
9
```

The terminal window below shows the execution of the program:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> javac q2c.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2> java q2c
Min Value: -128
Max Value: 127
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Assignments\Assignment 2>
```

- d. Declare a method-local variable number of type byte with some value and convert it to a String using the `toString` method.

The screenshot shows a Java code editor with a dark theme. A file named `q2d.java` is open, containing the following code:

```
1  q2d.java > ...
2  class q2d{
3      Run | Debug
4      public static void main(String[] args){
5          Byte number= 20;
6          String s= Byte.toString(number) ;
7          System.out.println(s);
8      }
9  }
10 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac q2d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java q2d
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> █
```

- e. Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method.

The screenshot shows a Java code editor with a dark theme. A file named q2e.java is open, containing the following code:

```
J q2e.java > ...
1  class q2e{
2      Run | Debug
3      public static void main(String[] args){
4          String strnumber= "20";
5          byte b= Byte.parseByte(strnumber) ;
6          System.out.println(b);
7      }
8  }
9
10
11
12
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q2e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q2e
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value.

The screenshot shows a Java code editor with a dark theme and a terminal window below it. The code in the editor is:

```
q2f.java > ...
1  class q2f{
2      Run | Debug
3      public static void main(String[] args){
4          String strnumber= "Ab12Cd3";
5          byte b= Byte.parseByte(strnumber) ;
6          System.out.println(b);
7      }
8  }
9
10
11
12
```

The terminal window shows the following output:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q2f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q2f
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:668)
        at java.base/java.lang.Byte.parseByte(Byte.java:193)
        at java.base/java.lang.Byte.parseByte(Byte.java:219)
        at q2f.main(q2f.java:5)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

The screenshot shows a Java code editor in VS Code with the following code:

```
J q2g.java > ...
1  class q2g{
2      Run | Debug
3      public static void main(String[] args){
4          Byte number= 20;
5          Byte.valueOf(number) ;
6          System.out.println(number);
7      }
8  }
9
10 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q2g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q2g
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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().

The screenshot shows a Java code editor with a dark theme. A file named q2h.java is open, containing the following code:

```
q2h.java > q2h
1  class q2h{
2      Run | Debug
3      public static void main(String[] args){
4          String strnumber= "30";
5          Byte.valueOf(strnumber) ;
6          System.out.println(strnumber);
7      }
8  }
9
10
```

The code defines a class q2h with a main method. Inside the main method, a String variable strnumber is assigned the value "30". The Byte.valueOf(strnumber) expression converts the string to a Byte object. Finally, System.out.println(strnumber) prints the string "30" to the console.

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac q2h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java q2h
30
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

The terminal output shows the command javac q2h.java to compile the Java file, followed by the command java q2h to run the program, resulting in the output 30.

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

The screenshot shows a Java code editor in VS Code with a dark theme. A file named `q2i.java` is open, containing the following code:

```
J q2i.java > q2i > main(String[])
1 public class q2i {
2
3     Run | Debug
4     public static void main(String[] args){
5         Byte b= 20;
6         System.out.println ((int)b);
7         System.out.println ((float)b);
8         System.out.println ((double)b);
9     }
10 }
11
12
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q2i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q2i
20
20.0
20.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

- a. Explore the Java API documentation for `java.lang.Short` 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.

The screenshot shows the Visual Studio Code interface with a Java code editor and a terminal window.

**Code Editor:**

```
q3b.java > ...
1  class q3b {
2
3      Run | Debug
4      public static void main(String[] args){
5          short a = Short.BYTES;
6          System.out.println(a);
7      }
8
9
10
11
12
```

**Terminal:**

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q3b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q3b
2
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

c. Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields.

The screenshot shows a Java code editor in VS Code. The code is as follows:

```
1  q3c.java > q3c > main(String[])
2
3  class q3c{
4      public static void main(String[] args){
5          short min = Short.MIN_VALUE;
6          Short max= Short.MAX_VALUE;
7          System.out.println(min);
8          System.out.println(max);
9      }
10 }
11
12
13
14
15
```

Below the code editor, there is a terminal window showing the execution of the program:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q3c.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q3c
-32768
32767
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

- d. Declare a method-local variable number of type short with some value and convert it to a String using the `toString` method

The screenshot shows a Java code editor with a dark theme. A file named `q3d.java` is open, containing the following code:

```
1  class q3d {  
2  
3      public static void main(String[] args){  
4          short number=10;  
5          String s = Short.toString(number);  
6          System.out.println(s);  
7      }  
8  }  
9  
10  
11  
12  
13
```

Below the code editor is a terminal window with the following history:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac q3d.java  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java q3d  
10  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

- e. Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method.

The screenshot shows a Java code editor with a dark theme. A file named Q3e.java is open, containing the following code:

```
J Q3e.java > ...
1 class Q3e{
2
3     public static void main(String[] args){
4         String strNumber = "10";
5         short a = Short.parseShort(strNumber);
6         System.out.println(a);
7     }
8 }
9
10
11
12
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q3e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q3e
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> []
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value.

The screenshot shows a Java code editor with a dark theme. A file named Q3f.java is open, containing the following code:

```
1 class Q3f{
2
3     Run | Debug
4     public static void main(String[] args){
5         String strNumber = "Ab12Cd3";
6         short a = Short.parseShort(strNumber);
7         System.out.println(a);
8     }
9 }
10
11
12
13
14
15
```

The code attempts to convert the string "Ab12Cd3" to a short integer using `Short.parseShort`. This results in a `NumberFormatException` because the string contains non-numeric characters. The terminal below shows the execution of the code and the resulting exception:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q3f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q3f
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:668)
        at java.base/java.lang.Short.parseShort(Short.java:137)
        at java.base/java.lang.Short.parseShort(Short.java:163)
        at Q3f.main(Q3f.java:6)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> 
```

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

```
J Q3g.java > ...
1
2     class Q3g{
3
4
5         Run | Debug
6         public static void main(String[] args){
7             short a = 15;
8             Short.valueOf(a);
9             System.out.println(a);
10            }
11
12
13
14
15
16
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q3g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q3g
15
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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().

The screenshot shows a Java code editor with a dark theme. A file named Q3h.java is open, containing the following code:

```
1  class Q3h{
2      public static void main(String args[])
3      {
4          String strNumber="14";
5          short a=Short.valueOf(strNumber);
6          System.out.println(a);
7      }
8 }
```

Below the code editor is a terminal window titled "powershell". The terminal output is as follows:

```
PS D:\C-DAC\OOPJ\Day 2\Programs> javac Q3h.java
PS D:\C-DAC\OOPJ\Day 2\Programs> java Q3h
14
PS D:\C-DAC\OOPJ\Day 2\Programs>
```



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

```
J Q3i.java > Q3i
1  class Q3i{
2
3      Run | Debug
4      public static void main(String[] args){
5          short a = 10;
6          System.out.println((int)a);
7          System.out.println((float)a);
8          System.out.println((long)a);
9          System.out.println((float)a);
10
11         int b=20;
12         float c=25.05f;
13         double d = 10.00;
14         System.out.println((short)b);
15         System.out.println((short)c);
16         System.out.println((short)d);
17     }
18 }
19
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q3i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q3i
10
10.0
10
10.0
20
25
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

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

- a. Explore the Java API documentation for `java.lang.Integer` 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.

```
J Q4b.java > ...
1  public class Q4b {
2
3      Run | Debug
4      public static void main(String[] args){
5          int a= Integer.BYTES;
6          System.out.println(a);
7      }
8  }
9
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4b
4
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> □
```

c. Write a program to find the minimum and maximum values of int using the MIN\_VALUE and MAX\_VALUE fields.

The screenshot shows a Java code editor in VS Code with the file 'Q4c.java' open. The code prints the minimum and maximum values of an integer. Below the editor is a terminal window showing the execution of the program.

```
J Q4c.java > ...
1  class Q4c {
2
3      Run | Debug
4          public static void main(String[] args){
5              int min = Integer.MIN_VALUE;
6              int max = Integer.MAX_VALUE;
7              System.out.println(min);
8              System.out.println(max);
9          }
10
11
12
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q4c.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q4c
-2147483648
2147483647
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

d. Declare a method-local variable number of type int with some value and convert it to a String using the `toString` method.

The screenshot shows a Java code editor window with a dark theme. The code in the editor is:

```
J Q4d.java > ...
1 public class Q4d {
2
3     Run | Debug
4     public static void main(String[] args){
5         int a = 10;
6         String s =Integer.toString(a);
7         System.out.println(s);
8     }
9
10 }
```

Below the editor is a terminal window showing the execution of the Java program:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4d
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

- e. Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method.

The screenshot shows a Java code editor window with a dark theme. A file named Q4e.java is open, containing the following code:

```
1 class Q4e{
2
3     public static void main(String[] args){
4         String s = "15";
5         int a=Integer.parseInt(s);
6         System.out.println(a);
7     }
8 }
```

Below the code editor is a terminal window showing the execution of the Java program:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4e
15
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

The terminal tabs at the bottom are labeled PROBLEMS (with a count of 1), OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined to indicate it is active), and PORTS.

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value.

The screenshot shows the Visual Studio Code interface with a dark theme. In the center is a code editor window displaying a Java file named Q4f.java. The code contains a class definition with a main method that attempts to parse a string into an integer. Below the code editor is a terminal window showing the execution of the Java program and its resulting error message.

```
J Q4f.java > Q4f > main(String[])
1
2  class Q4f{
3
4      Run | Debug
5      public static void main(String[] args){
6          String strNumber = "Ab12Cd3";
7          int a = Integer.parseInt(strNumber);
8          System.out.println(a);
9
10     }
11 }
12
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q4f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q4f
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:668)
        at java.base/java.lang.Integer.parseInt(Integer.java:786)
        at Q4f.main(Q4f.java:6)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

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

The screenshot shows a Java code editor with a dark theme. A code editor window displays the following Java code:

```
J Q4g.java > ...
1
2  class Q4g{
3  |
4      Run | Debug
5      public static void main(String[] args){
6          int a = 20;
7          Integer.valueOf(a);
8          System.out.println(a);
9
10     }
11 }
12
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4g
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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().

The screenshot shows the Visual Studio Code interface with a Java code editor and a terminal window.

**Code Editor:**

```
J Q4h.java > ...
1  class Q4h {
2
3      Run | Debug
4      public static void main(String[] args){
5          String strNumber = "15";
6          Integer.valueOf(strNumber);
7          System.out.println(strNumber);
8
9
10     }
11 }
12 }
13 }
```

**Terminal:**

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4h
15
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> █
```

- i. Declare two integer variables with values 10 and 20, and add them using a method from the Integer class.

The screenshot shows a Java code editor in VS Code with a dark theme. A file named Q4i.java is open, containing the following code:

```
J Q4i.java > Q4i > main(String[])
1 public class Q4i {
2
3     Run | Debug
4     public static void main(String[] args){
5         int a = 10;
6         int b= 20;
7         int res=Integer.sum(a,b);
8         System.out.println(res);
9     }
}
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4i
30
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

j. Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class.

```
J Q4j.java > ...
1  public class Q4j {
2      Run | Debug
3      public static void main(String[] args){
4          int a = 10;
5          int b= 20;
6          int min=Integer.min(a,b);
7          int max=Integer.max(a,b);
8          System.out.println("Minimum:" + min);
9          System.out.println("Maximum:" + max);
10 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4j
Minimum:10
Maximum:20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

k. Declare an integer variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Integer class.



The screenshot shows a Java code editor in VS Code with the following code:

```
J Q4k.java > Q4k
1 public class Q4k{
2
3     Run | Debug
4     public static void main(String[] args){
5         int a = 7;
6
7         System.out.println("Binary:" + Integer.toBinaryString(a));
8         System.out.println("Octal:" + Integer.toOctalString(a));
9         System.out.println("Hexadecimal:" + Integer.toHexString(a));
10    }
11 }
```

The code defines a class named Q4k with a main method. Inside the main method, an integer variable 'a' is assigned the value 7. Then, three println statements use the toBinaryString, toOctalString, and toHexString methods of the Integer class to print the binary, octal, and hexadecimal representations of 'a' respectively.

Below the code editor, the terminal tab is selected, showing the following command-line output:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4k.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4k
Binary:111
Octal:7
Hexadecimal:7
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

The screenshot shows a Java code editor with a dark theme. A file named Q4I.java is open, containing the following code:

```
Q4I.java > Q4I
1 public class Q4I{
2
3     Run | Debug
4     public static void main(String[] args){
5         int a = 10;
6         int b= 40;
7         float i =20.0f;
8         System.out.println((float)a);
9         System.out.println((double)a);
10        System.out.println((long)a);
11        System.out.println((float)b);
12        System.out.println((double)b);
13        System.out.println((long)b);
14        System.out.println((long)i);
15        System.out.println((double)i);
16        System.out.println((short)i);
17    }
}
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q4I.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q4I
10.0
10.0
10
40.0
40.0
40
20
20.0
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

- a. Explore the Java API documentation for `java.lang.Long` 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.

```
J Q5b.java > ...
1  public class Q5b {
2
3      Run | Debug
4      public static void main(String[] args){
5          long a = Long.BYTES;
6          System.out.println(a);
7      }
8
9 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PProgram vscode\Day1> javac Q5b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PProgram vscode\Day1> java Q5b
8
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PProgram vscode\Day1>
```

c. Write a program to find the minimum and maximum values of long using the MIN\_VALUE and MAX\_VALUE fields.

The screenshot shows a Java code editor with a dark theme. A file named Q5c.java is open, containing the following code:

```
Q5c.java > ...
1 class Q5c{
2     Run | Debug
3     public static void main(String[] args){
4         long min = Long.MIN_VALUE;
5         long max = Long.MAX_VALUE;
6         System.out.println(min);
7         System.out.println(max);
8     }
9 }
```

Below the code editor is a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5c.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5c
-9223372036854775808
9223372036854775807
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> []
```

- d. Declare a method-local variable number of type long with some value and convert it to a String using the `toString` method.

The screenshot shows a Java code editor with a dark theme. A file named `Q5d.java` is open, containing the following code:

```
J Q5d.java > ...
1  public class Q5d {
2
3
4      Run | Debug
5      public static void main(String[] args){
6          long number=10;
7          String s = Long.toString(number);
8          System.out.println(s);
9      }
10 }
11
12 
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5d
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> █
```

- e. Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method.

The screenshot shows the Visual Studio Code interface. The left pane displays a Java file named Q5e.java with the following code:

```
1 public class Q5e{  
2     public static void main(String[] args){  
3         string strNumber = "20";  
4         long a = Long.parseLong(strNumber);  
5         System.out.println(a);  
6     }  
7 }  
8 }
```

The right pane is a terminal window showing the execution of the Java program:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q5e.java  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q5e  
20  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value.

The screenshot shows a Java code editor with the file Q5f.java open. The code defines a class Q5f with a main method that attempts to parse the string "Ab12Cd3" into a long integer. The terminal below shows the execution of the code, resulting in a NumberFormatException due to the invalid input string.

```
J Q5f.java > ...
1 public class Q5f {
2
3     Run | Debug
4     public static void main(String[] args){
5         String strNumber = "Ab12Cd3";
6         long a = Long.parseLong(strNumber);
7         System.out.println(a);
8
9     }
10    }
11
12 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q5f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q5f
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 Q5f.main(Q5f.java:6)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

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

The screenshot shows a Java code editor in VS Code with the following code:

```
J Q5g.java > Q5g > main(String[])
1 public class Q5g {
2     Run | Debug
3         public static void main(String[] args){
4             long a = 15;
5             Long.valueOf(a);
6             System.out.println(a);
7
8         }
9     }
10
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q5g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q5g
15
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

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().

The screenshot shows a Java code editor with a dark theme. A file named Q5h.java is open, containing the following code:

```
1  Q5h.java > 🏃 Q5h > ⚙ main(String[])
2
3  public class Q5h {
4
5      public static void main(String[] args){
6
7          String strNumber = "14";
8          Long.valueOf(strNumber);
9          System.out.println(strNumber);
10
11     }
12 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5h
14
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

i. Declare two long variables with values 1123 and 9845, and add them using a method from the Long class.

The screenshot shows a Java code editor in VS Code with a dark theme. A file named `Q5i.java` is open, containing the following code:

```
J Q5i.java > Q5i > main(String[])
1 public class Q5i {
2
3     Run | Debug
4     public static void main(String[] args){
5         long a = 1123;
6         long b= 9845;
7         long res=Long.sum(a,b);
8         System.out.println(res);
9     }
10 }
```

The code uses the `Long.sum()` method to add two long integers and print the result. Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> javac Q5i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1> java Q5i
10968
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode\Day1>
```

j. Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class.

```
J Q5j.java > ...
1  public class Q5j {
2
3      Run | Debug
4      public static void main(String[] args){
5          long a = 1122;
6          long b= 5566;
7          long min=Long.min(a,b);
8          long max=Long.max(a,b);
9          System.out.println("Minimum:" + min);
10         System.out.println("Maximum:" + max);
11     }
12 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5j
Minimum:1122
Maximum:5566
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

k. Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class.

The screenshot shows the Visual Studio Code interface. At the top, there's a status bar with the text "J Q5k.java > ...". Below the status bar is a code editor window containing the following Java code:

```
1  public class Q5k {
2      Run | Debug
3      public static void main(String[] args){
4          long a = 7;
5          System.out.println("Binary:" + Long.toBinaryString(a));
6          System.out.println("Octal:" + Long.toOctalString(a));
7          System.out.println("Hexadecimal:" + Long.toHexString(a));
8      }
9 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q5k.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q5k
Binary:111
Octal:7
Hexadecimal:7
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

The screenshot shows a Java code editor and a terminal window. The code editor displays a file named Q51.java with the following content:

```
Q51.java > ...
1 public class Q51{
2
3     Run | Debug
4     public static void main(String[] args){
5         long a = 10;
6         long b= 40;
7         long c =20;
8         System.out.println((float)a);
9         System.out.println((double)a);
10        System.out.println((int)a);
11        System.out.println((float)b);
12        System.out.println((double)b);
13        System.out.println((int)b);
14        System.out.println((int)c);
15        System.out.println((double)c);
16        System.out.println((short)c);
17    }
18}
```

The terminal window below shows the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> javac Q51.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1> java Q51
10.0
10.0
10
40.0
40.0
40
20
20.0
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ PRogram vscode\Day1>
```

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

- a. Explore the Java API documentation for `java.lang.Float` 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.

J q6b.java X

J q6b.java > ...

```
1 class q6b{
2     Run | Debug
3     public static void main(String[] args){
4         float a =Float.BYTES;
5         System.out.println(a);
6     }
7 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6b.java

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6b

4.0

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>

c. Write a program to find the minimum and maximum values of float using the MIN\_VALUE and MAX\_VALUE fields. ss

The screenshot shows a Java code editor with a dark theme. A file named q6c.java is open, containing the following code:

```
J q6c.java  X
J q6c.java > ...
1 public class q6c {
2
3     Run | Debug
4     public static void main(String[] args){
5         float min =Float.MIN_VALUE;
6         float max =Float.MAX_VALUE;
7         System.out.println(min);
8         System.out.println(max);
9     }
10 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6c.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6c
1.4E-45
3.4028235E38
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> []
```

d. Declare a method-local variable number of type float with some value and convert it to a String using the `toString` method

```
J q6d.java > q6d
1 class q6d {
2
3     Run | Debug
4     public static void main(String[] args){
5         float number = 10f;
6         String s = Float.toString(number);
7         System.out.println(s);
8     }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6d
10.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

- e. Declare a method-local variable strNumber of type String with some value and convert it to a float value using the parseFloat method.

```
J q6e.java > ...
1 | class q6e {
2 |     Run|Debug
3 |     public static void main(String[] args){
4 |         String strNumber ="10";
5 |         float f = Float.parseFloat(strNumber);
6 |         System.out.println(f);
7 |
8 |     }
9 | }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6e
10.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a float value.

The screenshot shows a Java code editor with a dark theme and a terminal window below it. The code editor contains the following Java code:

```
J q6f.java > ...
1 public class q6f {
2     Run | Debug
3     public static void main(String[] args){
4         String strNumber = "Ab12Cd3";
5         float f = Float.parseFloat(strNumber);
6         System.out.println(f);
7     }
8 }
9
10
```

The terminal window shows the following command-line interaction:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6f
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:476)
        at q6f.main(q6f.java:4)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

g. Declare a method-local variable number of type float with some value and convert it to the corresponding wrapper class using Float.valueOf().

Day1 > Day2 > J q6g.java > ...

```
1
2     class q6g{
3
4         Run | Debug
5         public static void main(String[] args){
6             float i = 20;
7             Float.valueOf(i);
8             System.out.println(i);
9
10        }
11    }
12
13
14
15
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6g
20.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

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().

```
J q6h.java > ...
1  public class q6h {
2
3      Run | Debug
4      public static void main(String[] args){
5          String strNumber = "1020";
6          float f = Float.valueOf(strNumber);
7          System.out.println(f);
8
9      }
10
11
12
13
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6h
1020.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

- i. Declare two float variables with values 112.3 and 984.5, and add them using a method from the Float class.

The screenshot shows the Visual Studio Code interface. The top half displays the code editor with a dark theme, containing the following Java code:

```
J q6i.java > ...
1
2     class q6i {
3         Run | Debug
4             public static void main(String[] args){
5                 float a =112.3f;
6                 float b= 984.5f;
7                 float res=Float.sum(a,b);
8                 System.out.println(res);
9
10
11
12 }
13
14
```

The bottom half shows the terminal window with the following output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6i
1096.8
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

j. Declare two float variables with values 112.2 and 556.6, and find the minimum and maximum values using the Float class.

```
J q6j.java > ...
1
2     class q6j{
3         Run | Debug
4         public static void main(String[] args){
5             float a =112.2f;
6             float b= 556.6f;
7             float min = Float.min(a,b);
8             float max= Float.max(a,b);
9             System.out.println("Minimum: "+ min);
10            System.out.println("Maximum: "+ max);
11
12        }
13    }
14
15 }
16
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6j
Minimum: 112.2
Maximum: 556.6
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

k. Declare a float variable with the value -25.0f. Find the square root of this value.

```
J q6k.java > ...
1  class q6k {
2      Run | Debug
3      public static void main(String[] args){
4          float a =-25.0f;
5          System.out.println(Math.sqrt(a));
6      }
7  }
8
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6k.java

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6k

NaN

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>

1. Declare two float variables with the same value, 0.0f, and divide them

The screenshot shows a Java code editor in VS Code with a dark theme. A file named `q6l.java` is open, containing the following code:

```
1  q6l.java > q6l > main(String[])
2
3  class q6l {
4      Run | Debug
5      public static void main(String[] args){
6          float a=0.0f;
7          float b=0.0f;
8          System.out.println(a/b);
9      }
10 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6l.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6l
NaN
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

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

The screenshot shows a Java code editor with a dark theme. A file named q6m.java is open, containing the following code:

```
J q6m.java > ...
1 class q6m {
2
3     Run | Debug
4     public static void main(String[] args){
5         float a=10.5f;
6         System.out.println((int)a);
7         System.out.println((float)a);
8         System.out.println((double)a);
9         System.out.println((long)a);
10
11         float d=a;
12         short s =(short) a;
13         System.out.println(d);
14         System.out.println(s);
15
16     }
17 }
18
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q6m.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q6m
10
10.5
10.5
10
10.5
10
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

## 7. Working with java.lang.Double

- a. Explore the Java API documentation for java.lang.Double 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.

The screenshot shows a Java code editor with a dark theme. A file named `q7b.java` is open, containing the following code:

```
J q7b.java > ...
1  class q7b {
2      Run | Debug
3      public static void main(String[] args){
4          double a =Double.BYTES;
5          System.out.println(a);
6
7
8      }
9
10 }
11
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7b.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7b
8.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

c. Write a program to find the minimum and maximum values of double using the MIN\_VALUE and MAX\_VALUE fields.

The screenshot shows the Visual Studio Code interface. In the top-left, there's a file tab labeled "q7c.java". Below it is the code editor with the following Java code:

```
1 class q7c {  
2     public static void main(String[] args){  
3         Double min =Double.MIN_VALUE;  
4         Double max =Double.MAX_VALUE;  
5         System.out.println(min);  
6         System.out.println(max);  
7     }  
8 }  
9
```

Below the code editor, the interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is currently selected, showing the following terminal output:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7c.java  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7c  
4.9E-324  
4.9E-324  
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

- d. Declare a method-local variable number of type double with some value and convert it to a String using the `toString` method.

The screenshot shows a Java code editor window with a dark theme. A file named `q7d.java` is open, containing the following code:

```
1 class q7d {
2     Run | Debug
3     public static void main(String[] args){
4         Double d = 20.008;
5         String str =Double.toString(d);
6         System.out.println(str);
7     }
8 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7d.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7d
20.008
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

The terminal tabs at the bottom are labeled: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

- e. Declare a method-local variable strNumber of type String with some value and convert it to a double value using the parseDouble method.

```
J q7e.java > ...
1  class q7e {
2
3      Run | Debug
4      public static void main(String[] args){
5          String strNumber="20";
6          Double d=Double.parseDouble(strNumber);
7          System.out.println(d);
8      }
9 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7e.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7e
20.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> █
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a double value.

```
J q7f.java > ...
1  class q7f {
2
3
Run|Debug
4  public static void main(String[] args){
5    String strNumber="Ab12Cd3";
6    Double d=Double.parseDouble(strNumber);
7    System.out.println(d);
8  }
9 }
10
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7f.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7f
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:651)
        at q7f.main(q7f.java:6)
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

g. Declare a method-local variable number of type double with some value and convert it to the corresponding wrapper class using Double.valueOf().

```
J q7g.java > ...
1  class q7g {
2      Run | Debug
3      public static void main(String[] args){
4          Double d= 20.08;
5          Double.valueOf(d);
6          System.out.println(d);
7      }
8  }
9
10
11
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7g.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7g
20.08
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

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().

The screenshot shows a Java code editor window with a dark theme. A file named q7h.java is open, containing the following code:

```
J q7h.java > ...
1  class q7h {
2
3      public static void main(String[] args){
4
5          String strNumber="4078";
6          double i= Double.valueOf(strNumber);
7          System.out.println(i);
8      }
9
10 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7h.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7h
4078.0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

- i. Declare two double variables with values 112.3 and 984.5, and add them using a method from the Double class.

The screenshot shows the Visual Studio Code interface. The top half displays a Java code editor with the following content:

```
J q7i.java > q7i > main(String[])
1  class q7i {
2      Run | Debug
3      public static void main(String[] args){
4          double a= 112.3 ;
5          double b=984.5;
6          double res= Double.sum(a,b);
7          System.out.println(res);
8      }
9  }
10 }
11
```

The bottom half shows a terminal window with the following command-line session:

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7i.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7i
1096.8
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

j. Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class.

```
J q7j.java > ...
1  | class q7j {
2  |     Run | Debug
3  |     public static void main(String[] args){
4  |         double a= 112.2 ;
5  |         double b=556.6;
6  |         double min=Double.min(a,b);
7  |         double max=Double.max(a,b);
8  |         System.out.println("Minimum: " + min);
9  |         System.out.println("Maximum: " +max);
10 |
11 |
12     }
13
14 }
15
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7j.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7j
Minimum: 112.2
Maximum: 556.6
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

k. Declare a double variable with the value -25.0. Find the square root of this value.

```
J q7k.java > ...
1  | class q7k {
2  |     Run | Debug
3  |     public static void main(String[] args){
4  |
5  |         Double a= -25.0;
6  |
7  |         System.out.println(Math.sqrt(a));
8  |
9  |
10 |
11    }
12
13  }
14
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7k.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7k
NaN
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

1. Declare two double variables with the same value, 0.0, and divide them.

```
J q71.java > ...
1 | class q71 {
|   Run | Debug
2 |   public static void main(String[] args){
3 |
4 |
5 |     Double a= 0.0;
6 |     Double b=0.0;
7 |
8 |     System.out.println(a/b);
9 |
10|
11|
12|
13| }
14|
15|
16|
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q71.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q71
NaN
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

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

```
J q7m.java > ...
1  class q7m {
2      public static void main(String[] args){
5          double a= 20.8;
6          System.out.println((float)a);
7          System.out.println((int)a);
8          System.out.println((long)a);
9          System.out.println((short)a);
10
11         int d=(int)a;
12         short s =(short)a;
13         System.out.println(d);
14         System.out.println(s);
15     }
16
17 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac q7m.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java q7m
20.8
20
20
20
20
20
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

## 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()`).

```
J Q8.java > ...
1
2 import java.util.*;
3 class Q8{
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner sc= new Scanner(System.in);
7         System.out.println("enter byte: ");
8         byte b=sc.nextByte();
9         System.out.println("enter short: ");
10        short s=sc.nextShort();
11        System.out.println("enter integer: ");
12        int i=sc.nextInt();
13        System.out.println("enter float: ");
14        float f=sc.nextFloat();
15        System.out.println("enter long: ");
16        long l=sc.nextLong();
17        System.out.println("enter double: ");
18        double d=sc.nextDouble();
19
20        String s1=Byte.toString(b);
21        String s2=Short.toString(s);
22        String s3=Integer.toString(i);
23        String s4=Float.toString(f);
24        String s5=Long.toString(l);
25        String s6=Double.toString(d);
26
27        System.out.println( s1 +"" + s2 + "" + s3 + "" + s4 + "" + s5 + "" + s6 );
```

```
String a1=String.valueOf(b);
String a2=String.valueOf(s);
String a3=String.valueOf(i);
String a4=String.valueOf(f);
String a5=String.valueOf(l);
String a6=String.valueOf(d);
System.out.println( a1 + " " + a2+ " " + a3 + " " + a4+ " " + a5+ " " + a6 );
sc.close();
}
```

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac Q8.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java Q8
enter byte:
1
enter short:
4
enter integer:
20
enter float:
16.4
enter long:
16
```

```
enter double:
20.8
142016.41620.8
142016.41620.8
```

## **9. Default Values of Primitive Types**

Declare variables of each primitive type as fields of a class and check their default values.

```
J Q9.java > ...
1  class Q9 {
2      short s;
3      int a;
4      float f;
5      double d;
6      long l;
7      Run | Debug
8      public static void main (String[] args){
9          Q9 obj= new Q9();
10         System.out.println(obj.s);
11         System.out.println(obj.a);
12         System.out.println(obj.f);
13         System.out.println(obj.d);
14         System.out.println(obj.l);
15     }
16
17 }
18
19
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> javac Q9.java
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode> java Q9
0
0
0.0
0.0
0
PS C:\Users\swapn\OneDrive\Desktop\OOPJ Program vscode>
```

## 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.

```
class Q10{  
    public static void main(String args[]){  
        if(args.length != 3){  
            System.out.println("Usage: java ArithmeticOperations <num1>  
<operator> <num2>");  
            return;  
        }  
  
        try {  
            int num1 = Integer.parseInt(args[0]);  
            String operator = args[1];  
            int num2 = Integer.parseInt(args[2]);  
            int result;  
  
            switch(operator){  
                case "+":  
                    result = num1 + num2;  
                    break;  
                case "-":  
                    result = num1 - num2;  
                    break;  
                case "*":  
                    result = num1 * num2;  
            }  
        } catch (Exception e) {  
            System.out.println("Error: Invalid input");  
        }  
    }  
}
```

```

        break;

    case "/":
        if (num2 == 0) {
            System.out.println("Error: Division by zero is not allowed.");
            return;
        }
        result = num1 / num2;
        break;

    default:
        System.out.println("Error: Invalid operator. Use +, -, *, or /.");
        return;
    }

    System.out.println("Result: " + result);
} catch (NumberFormatException e) {
    System.out.println("Error: Please provide valid integers.");
}
}
}

```

```

PS D:\C-DAC\OOPJ\Day 2\Programs> javac Q10.java
PS D:\C-DAC\OOPJ\Day 2\Programs> java Q10
Usage: java ArithmeticOperations <num1> <operator> <num2>
PS D:\C-DAC\OOPJ\Day 2\Programs> 10 * 20
200
PS D:\C-DAC\OOPJ\Day 2\Programs> █

```