**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]() and observe its modifiers and super types.

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

***Field Summary***

**Fields**

| Modifier and Type | Field and Description |
|---|---|
| `static Boolean` | `FALSE`<br>The `Boolean` object corresponding to the primitive value `false`. |
| `static Boolean` | `TRUE`<br>The `Boolean` object corresponding to the primitive value `true`. |
| `static Class<Boolean>` | `TYPE`<br>The Class object representing the primitive type boolean. |

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

```java
package Assgn_3;

public class A3_1 {

    public static void main(String[] args) {
     {
            boolean status = true;
            String statusString = Boolean.toString(status);
            System.out.println(statusString);
        }

    }
}
```

```
<terminated> A3_1 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (06-Se
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)).

```java
package Assgn_3;

public class A3_3 {

    public static void main(String[] args) {
        String strStatus = "1";
        boolean status = Boolean.parseBoolean(strStatus);
        System.out.println(status);

    }

}
```

```
<terminated> A3_3 [Java Application] C:\Program Files\Java\jdk-21\bi
false
```

**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").

```java
package Assgn_3;

public class A3_4 {

    public static void main(String[] args) {
        boolean status = true;
        Boolean statusWrapper = Boolean.valueOf(status);
        System.out.println(statusWrapper);
    }

}
```

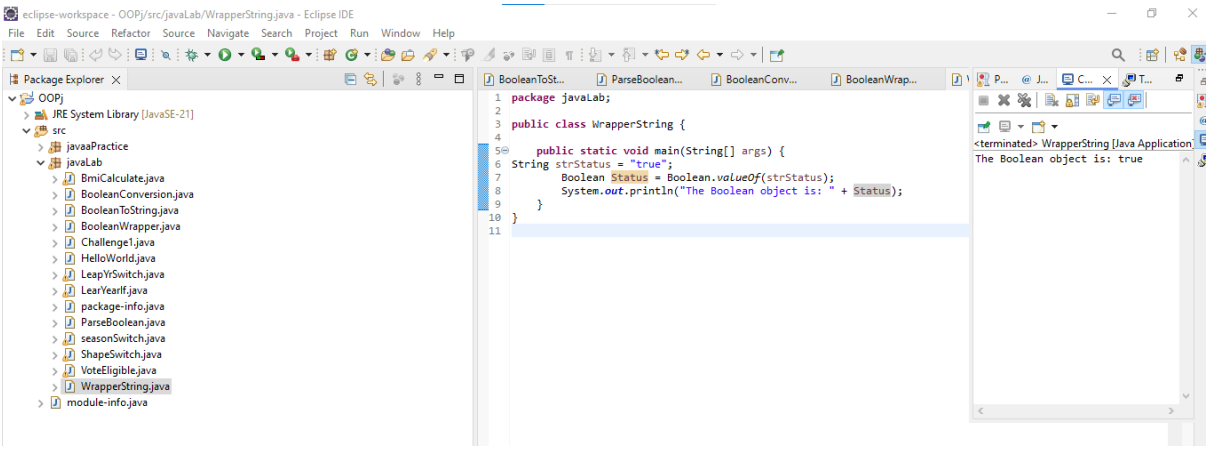
```
<terminated> A3_4 [Java Application] C:\Proc
true
```

**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)).

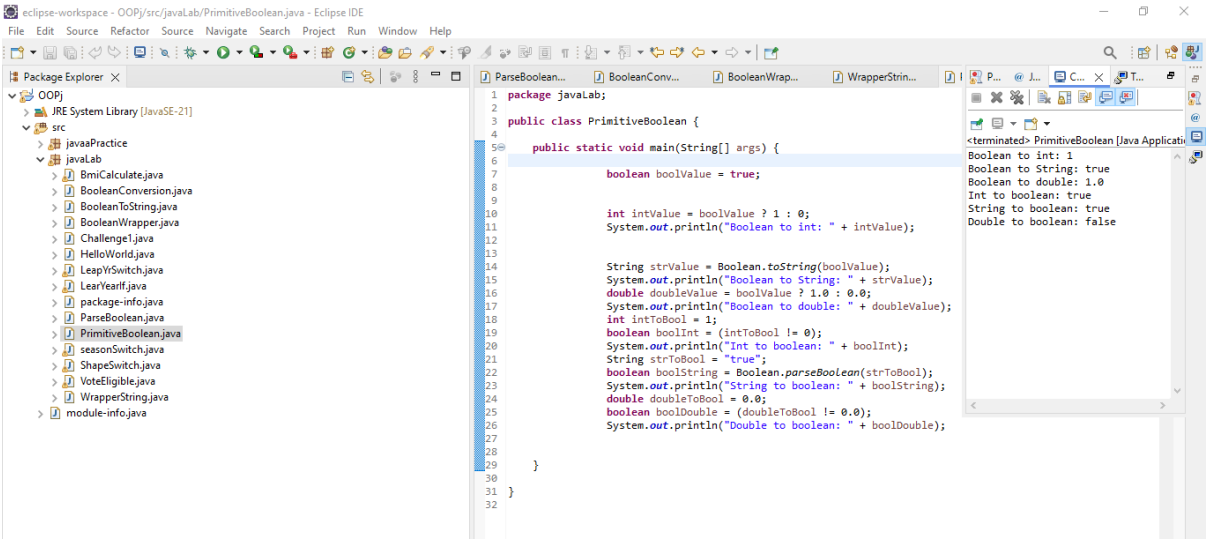
```java
package javaLab;

public class BooleanConversion {

    public static void main(String[] args) {
        String strStatus = "1";
        boolean boolStatus = convertToBoolean(strStatus);
        System.out.println("The boolean value is: " + boolStatus);
    }

    public static boolean convertToBoolean(String str) {
        return "1".equals(str);
    }
}
```

```
<terminated> BooleanConversion [Java Application] C
The boolean value is: 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)).




**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]() and observe its modifiers and super types.

### *Field Summary*

**Fields**

| Modifier and Type | Field and Description |
|---|---|
| static int | **BYTES**<br>The number of bytes used to represent a byte value in two's complement binary form. |
| static byte | **MAX_VALUE**<br>A constant holding the maximum value a byte can have, $2^7$-1. |
| static byte | **MIN_VALUE**<br>A constant holding the minimum value a byte can have, -$2^7$. |
| static int | **SIZE**<br>The number of bits used to represent a byte value in two's complement binary form. |
| static Class<Byte> | **TYPE**<br>The Class instance representing the primitive type byte. |

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

```java
package javaLab;

public class SizeByte {

    public static void main(String[] args) {
                int numberOfBytes = Byte.BYTES;
                System.out.println(" byte value is: " + numberOfBytes);
            }
}
```

```
<terminated> SizeByte [Java A
 byte value is: 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).

```java
package javaLab;

public class StringByte {

    public static void main(String[] args) {
        byte number = 42;
            String numberAsString = Byte.toString(number);
            System.out.println("The byte value as a String is: " + numberAsString);

    }

}
```

```
<terminated> StringByte [Java Application] C:\Eclips
The byte value as a String is: 42
```

**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)).

```java
package Assgn_3;

public class A3_2_1 {

    public static void main(String[] args) {
        byte number = 42;
        String numberAsString = Byte.toString(number);
        System.out.println("The byte value as a String is: " + numberAsString);
    }

}
```

```
<terminated> A3_2_1 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
The byte value as a String is: 42
```

**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)).

```java
package javaLab;

public class ByteString {

    public static void main(String[] args) {
        String strNumber = "50";
            byte number = Byte.parseByte(strNumber);
            System.out.println("The String value as a byte is: " + number);
    }

}
```

```
<terminated> ByteString [Java Application]
The String value as a byte is: 50
```

**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).

```
<terminated> ByteException [Java Application] C:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk
Exception in thread "main" java.lang.NumberFormatException: For input st
        at java.base/java.lang.NumberFormatException.forInputString(Numb
        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 OOPj/javaLab.ByteException.main(ByteException.java:7)
```

```java
package javaLab;

public class ByteException {

    public static void main(String[] args) throws NumberFormatException  {
        String strNumber = "Ab12Cd3";
        byte number = Byte.parseByte(strNumber);
        System.out.println("The String value as a byte is: " + number);
    }

}
```

**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)).

```
<terminated> ByteValue [Java Application] C:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hots
 Byte wrapper object is: 10
```

```java
package javaLab;

public class ByteValue {

    public static void main(String[] args) {
        byte number = 10;
          Byte byteWrapper = Byte.valueOf(number);
          System.out.println(" Byte wrapper object is: " + byteWrapper);
    }

}
```

**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)).

```
<terminated> ByteStringConvert [Java Application] C:\Eclipse\eclipse\plugins\org.eclipse.justj.oper
The String value as a Byte wrapper object is: 85
```

```java
package javaLab;

public class ByteStringConvert {

    public static void main(String[] args) {
        String strNumber = "85";
         Byte byteWrapper = Byte.valueOf(strNumber);
         System.out.println("The String value as a Byte wrapper object is: " + byteWrapper);
    }

}
```

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

```
<terminated> ConversionByte [Java Application] C:\Eclipse\eclipse\plugins\org.eclipse.justj.openjd
byte value: 70
Converted to int: 70
Converted to short: 70
Converted to long: 70
Converted to float: 70.0
Converted to double: 70.0
Converted to char: F
Converted int to byte: 127
Converted double to byte: 42
```

```java
package javaLab;

public class ConversionByte {

    public static void main(String[] args) {
        byte byteValue = 70;
        int intValue = byteValue;
        short shortValue = byteValue;
        long longValue = byteValue;
        float floatValue = byteValue;
        double doubleValue = byteValue;
        char charValue = (char) byteValue;
        System.out.println("byte value: " + byteValue);
        System.out.println("Converted to int: " + intValue);
        System.out.println("Converted to short: " + shortValue);
        System.out.println("Converted to long: " + longValue);
        System.out.println("Converted to float: " + floatValue);
        System.out.println("Converted to double: " + doubleValue);
        System.out.println("Converted to char: " + charValue);
        int intValue2 = 127;
        byte byteValueFromInt = (byte) intValue2;
        System.out.println("Converted int to byte: " + byteValueFromInt);
        double doubleValue2 = 42.5;
        byte byteValueFromDouble = (byte) doubleValue2;
        System.out.println("Converted double to byte: " + byteValueFromDouble);

    }
```

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

**a.** Explore the [Java API documentation for java.lang.Short]() and observe its modifiers and super types.

**Fields**

| Modifier and Type | Field and Description |
|---|---|
| `static int` | **BYTES**<br>The number of bytes used to represent a `short` value in two's complement binary form. |
| `static short` | **MAX_VALUE**<br>A constant holding the maximum value a `short` can have, $2^{15}$-1. |
| `static short` | **MIN_VALUE**<br>A constant holding the minimum value a `short` can have, -$2^{15}$. |
| `static int` | **SIZE**<br>The number of bits used to represent a `short` value in two's complement binary form. |
| `static Class<Short>` | **TYPE**<br>The `Class` instance representing the primitive type `short`. |

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

```java
package Assgn_3;

public class A3_2_e
{
    public static void main(String[] args)
    {
        /*signed short  test  ;
         test=-17;
         System.out.println(" -17 when viewed as signed short Hexa is \t %x\n "+ test);*/
        int bytesUsed = Short.BYTES;
        System.out.println("The number of bytes used to represent a short value is: " + bytesUsed);
    }
}
```

```
<terminated> A3_2_e [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
The number of bytes used to represent a short value is: 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).

```java
package Assgn_3;

public class A3_3_c
{
    public static void main(String[] args)
    {
        short minValue = Short.MIN_VALUE;
        short maxValue = Short.MAX_VALUE;

        System.out.println("The minimum value of a short is: " + minValue);
        System.out.println("The maximum value of a short is: " + maxValue);

    }

}
```

```
<terminated> A3_3_c [Java Application] C:\Program Files\Java\jdk-
The minimum value of a short is: -32768
The maximum value of a short is: 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)).

```java
package Assgn_3;

public class A3_3_d
{

    public static void main(String[] args)
    {
        short number = 12345;
            String numberAsString = Short.toString(number);
            System.out.println("The short value as a String is: " + numberAsString);

    }

}
```

```
<terminated> A3_3_d [Java Application] C:\Program Files\
The short value as a String is: 12345
```

**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)).

```java
package Assgn_3;

public class A3_3_e
{
    public static void main(String[] args)
    {
        String strNumber = "5678";
        short number = Short.parseShort(strNumber);
        System.out.println("The short value is: " + number);

    }

}
```

```
<terminated> A3_3_e [Java Applicati
The short value is: 5678
```

**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).

```java
package Assgn_3;

public class A3_3_f
{
            public static void main(String[] args) {
            String strNumber = "Ap23pd1";
                try
                {
                short number = Short.parseShort(strNumber);
                System.out.println("The short value is: " + number);
                }
                catch (NumberFormatException e) {
                System.out.println("NumberFormatException occurred: " + e.getMessage());
            }
        }
}
```

```
<terminated> A3_3_f [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (06-Se
NumberFormatException occurred: For input string: "Ap23pd1"
```

**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)).

A3_3_g.java

```java
package Assgn_3;

public class A3_3_g
{
    public static void main(String[] args)
    {
        short number = 6789;
            Short wrapperNumber = Short.valueOf(number);
            System.out.println("The Short object is: " + wrapperNumber);
    }

}
```

Problems Javadoc Declaratic

```
<terminated> A3_3_g [Java Application]
The Short object is: 6789
```

**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)).

A3_3_g.java A3_3_h.java

```java
package Assgn_3;

public class A3_3_h
{
    public static void main(String[] args)
    {
        String strNumber = "6789";
        Short wrapperNumber = Short.valueOf(strNumber);
        System.out.println("The Short object from String is: " + wrapperNumber);

    }

}
```

Problems Javadoc Declaration Console

```
<terminated> A3_3_h [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (06-Sept
The Short object from String is: 6789
```

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

A3_3_g.java A3_3_h.java A3_3_i.java

```java
package Assgn_3;

public class A3_3_i
{

    public static void main(String[] args)
    {
        short shortValue = 400;
        int intValue = shortValue;
        long longValue = shortValue;
        float floatValue = shortValue;
        double doubleValue = shortValue;
        System.out.println("short to int:" +intValue);
        System.out.println("short to long:" +longValue);
        System.out.println("short to float:" +floatValue);
        System.out.println("short to double:" +doubleValue);
        int largeInt = 900;
        short shortFromInt = (short) largeInt;
        byte byteFromInt = (byte) largeInt;
        System.out.println("int to short (with possible data loss): "+ shortFromInt);
        System.out.println("int to byte (with possible data loss): " + byteFromInt);
        byte byteValue = 50;
        short shortFromByte = byteValue;
        System.out.println("byte to short:" +shortFromByte);
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> A3_3_i [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
short to int:400
short to long:400
short to float:400.0
short to double:400.0
int to short (with possible data loss): 900
int to byte (with possible data loss): -124
byte to short:50
```

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

**a.** Explore the [Java API documentation for java.lang.Integer]() and observe its modifiers and super types.

**Fields**

| Modifier and Type | Field and Description |
|---|---|
| `static int` | **BYTES**<br>The number of bytes used to represent a `long` value in two's complement binary form. |
| `static long` | **MAX_VALUE**<br>A constant holding the maximum value a `long` can have, $2^{63}$-1. |
| `static long` | **MIN_VALUE**<br>A constant holding the minimum value a `long` can have, -$2^{63}$. |
| `static int` | **SIZE**<br>The number of bits used to represent a `long` value in two's complement binary form. |
| `static Class<Long>` | **TYPE**<br>The `Class` instance representing the primitive type `long`. |

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

```
package Assgn_3;

public class A3_4_b {
        public static void main(String[] args) {
                int bytesUsed = Integer.BYTES;
    System.out.println("Number of bytes used to represent an int: " + bytesUsed);
        }

}
```

Output: Number of bytes used to represent an int: 4

**c.** Write a program to find the minimum and maximum values of int using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Integer.MIN_VALUE and Integer.MAX_VALUE).

```
package Assgn_3;

public class A3_4_c {

public static void main(String[] args) {
    int minValue = Integer.MIN_VALUE;
    int maxValue = Integer.MAX_VALUE;
    System.out.println("Minimum value of an int: " + minValue);
    System.out.println("Maximum value of an int: " + maxValue);
        }

}
```

Output: Minimum value of an int: -2147483648

Maximum value of an int: 2147483647

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

```
package Assgn_3;
public class A3_4_d {
                public static void main(String[] arg)
        {
                        System.out.println("Integer.MAX_VALUE = "
                          + Integer.MAX_VALUE);
        }
      }
```

**e.** Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method. (Hint: Use Integer.parseInt(String)).

```
package Assgn_3;

public class A3_4_e {

        public static void main(String[] args) {

                int a=1234;

                int b=-1234;

                 String str1 = Integer.toString(a);

            String str2 = Integer.toString(b);

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

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

        }


}
```

Output: String str1 = 1234

String str2 = -1234

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

```
package Assgn_3;
public class A3_4_f {
        public static void main(String[] args) {
                String numberStr = "12345";
    int number = Integer.parseInt(numberStr);
    System.out.println("The converted number is: " + number);
        }
}
```

Output: The converted number is: 12345

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

```
package Assgn_3;
public class A3_4_g {
        public static void main(String[] args) {
                String strNumber = "Ap27Pd16";
    try {
      int number = Integer.parseInt(strNumber);
      System.out.println("Converted number: " + number);
    } catch (NumberFormatException e) {
      System.out.println("Error: Invalid number format.");
          }
        }
}
```

Output: Error: Invalid number format.

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

```
package Assgn_3;
public class A3_4_h {
        public static void main(String[] args) {
                int number = 95;
    Integer integerObject = Integer.valueOf(number);
    System.out.println("Integer object: " + integerObject);
        }
}
```

Output: Integer object: 95

**i.** Declare two integer variables with values 10 and 20, and add them using a method from the Integer class. (Hint: Use Integer.sum(int, int)).

```
package Assgn_3;
public class A3_4_i1 {
        public static void main(String[] args) {
                int num1 = 50;
    int num2 = 20;
    int sum = Integer.sum(num1, num2);
    System.out.println("The sum of " + num1 + " and " + num2 + " is " + sum);
        }
}
```

Output: The sum of 50 and 20 is 70

**j.** Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class. (Hint: Use Integer.min(int, int) and Integer.max(int, int)).

```
package Assgn_3;
public class A3_4_j {
```

```
        public static void main(String[] args) {
                int num1 = 50;
    int num2 = 90;
    int minValue = Integer.min(num1, num2);
    int maxValue = Integer.max(num1, num2);
    System.out.println("The minimum value between " + num1 + " and " + num2 + " is " +
minValue);
    System.out.println("The maximum value between " + num1 + " and " + num2 + " is " +
maxValue);
  }
}
```

Output: The minimum value between 50 and 90 is 50

The maximum value between 50 and 90 is 90

**k.** Declare an integer variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Integer class. (Hint: Use Integer.toBinaryString(int), Integer.toOctalString(int), and Integer.toHexString(int)).

```
package Assgn_3;
public class A3_4_k {
        public static void main(String[] args) {
                int number = 9;
    String binaryString = Integer.toBinaryString(number);
    String octalString = Integer.toOctalString(number);
    String hexString = Integer.toHexString(number);
    System.out.println("Binary representation of " +number + " is " + binaryString);
    System.out.println("Octal representation of " +number + " is " + octalString);
    System.out.println("Hexadecimal representation of " +number+" is "+ hexString);
  }
}
```

Output: Binary representation of 9 is 1001

Octal representation of 9 is 11
Hexadecimal representation of 9 is 9

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

```
package Assgn_3;

public class A3_4_l {

        public static void main(String[] args) {

                int intValue=300;
                byte byteValue = (byte) intValue;
                System.out.println("Byte value: " + byteValue);

                short shortValue = (short) intValue;
                System.out.println("Short value: " + shortValue);
```

```
        long longValue = (long) intValue;
        System.out.println("Long value: " + longValue);

        float floatValue = (float) intValue;
        System.out.println("Float value: " + floatValue);

        double doubleValue = (double) intValue;
        System.out.println("Double value: " + doubleValue);
    }
}
```

Output: Byte value: 44
Short value: 300
Long value: 300
Float value: 300.0
Double value: 300.0

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

**a.** Explore the [Java API documentation for java.lang.Long]() and observe its modifiers and super types.

**Fields**

| Modifier and Type | Field and Description |
|---|---|
| `static int` | **BYTES**<br>The number of bytes used to represent a `long` value in two's complement binary form |
| `static long` | **MAX_VALUE**<br>A constant holding the maximum value a `long` can have, $2^{63}$-1. |
| `static long` | **MIN_VALUE**<br>A constant holding the minimum value a `long` can have, -$2^{63}$. |
| `static int` | **SIZE**<br>The number of bits used to represent a `long` value in two's complement binary form. |
| `static Class<Long>` | **TYPE**<br>The `Class` instance representing the primitive type `long`. |

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

```
package Assgn_3;
public class A3_5_b {
        public static void main(String[] args) {
  int bytesUsed = Long.BYTES;
  System.out.println("The number of bytes used to represent a long value: " + bytesUsed);
        }
}
```

Output: The number of bytes used to represent a long value: 8

**c.** Write a program to find the minimum and maximum values of long using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Long.MIN_VALUE and Long.MAX_VALUE).

```
package Assgn_3;
public class A3_5_c {
        public static void main(String[] args) {
  long minValue = Long.MIN_VALUE;
  long maxValue = Long.MAX_VALUE;
  System.out.println("The minimum value of a long is: " + minValue);
  System.out.println("The maximum value of a long is: " + maxValue);
        }
}
```

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

The maximum value of a long is: 9223372036854775807

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

```
package Assgn_3;
public class A3_5_d {
        public static void main(String[] args) {
  long number = 123456789L;
   String numAsString = Long.toString(number);
    System.out.println("The long value as a String is: " + numAsString);
        }
}
```

Output: The long value as a String is: 123456789

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

```
package Assgn_3;
public class A3_5_e {
        public static void main(String[] args) {
                String strNumber = "2334445960";
    long number = Long.parseLong(strNumber);
    System.out.println("The String value as a long is: " + number);
        }
}
```

Output: The String value as a long is: 2334445960

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

```
package Assgn_3;
public class A3_5_f {
        public static void main(String[] args) {
  String strNumber = "Ab12Cd3";
  long number = Long.parseLong(strNumber);
  System.out.println("The String value as a long is: " + number);
```

```
}
}
```
Output: Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
```
	at
java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
	at java.base/java.lang.Long.parseLong(Long.java:709)
	at java.base/java.lang.Long.parseLong(Long.java:832)
	at Assgn_3.A3_5_f.main(A3_5_f.java:5)
```

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

```
package Assgn_3;

public class A3_5_g {

	public static void main(String[] args) {

  long number = 239375053L;

  Long numberWrapper = Long.valueOf(number);

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

	}

}
```

Output: The Long object is: 239375053

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

```
package Assgn_3;
public class A3_5_h {
	public static void main(String[] args) {
		String strNumber = "1234567890";
		Long longValue = Long.valueOf(strNumber);
    System.out.println("The Long value is: " + longValue);
	}
}
```
Output: The Long value is: 1234567890

**i.** Declare two long variables with values 1123 and 9845, and add them using a method from the Long class. (Hint: Use Long.sum(long, long)).

```
package Assgn_3;
public class A3_5_i {
```

```
        public static void main(String[] args) {
                long num1 = 2020;
    long num2 = 2025;
    long sum = Long.sum(num1, num2);
    System.out.println("The sum of " + num1 + " and " + num2 + " is: " + sum);
        }
}
```

Output : The sum of 2020 and 2025 is: 4045

**j.** Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class. (Hint: Use Long.min(long, long) and Long.max(long, long)).

```
package Assgn_3;
public class A3_5_j {
        public static void main(String[] args) {
                long num1 = 1122;
    long num2 = 5566;
    long minValue = Long.min(num1, num2);
    long maxValue = Long.max(num1, num2);
    System.out.println("The minimum value between " + num1 + " and " + num2 + " is: " +
minValue);
    System.out.println("The maximum value between " + num1 + " and " + num2 + " is: " +
maxValue);
  }
}
```

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

The maximum value between 1122 and 5566 is: 5566

**k.** Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class. (Hint: Use Long.toBinaryString(long), Long.toOctalString(long), and Long.toHexString(long)).

```
package Assgn_3;
public class A3_5_k {
        public static void main(String[] args) {
                long number = 7;
    String binaryString = Long.toBinaryString(number);
    String octalString = Long.toOctalString(number);
    String hexString = Long.toHexString(number);
    System.out.println("Binary representation of " + number + " is: " + binaryString);
    System.out.println("Octal representation of " + number + " is: " + octalString);
    System.out.println("Hexadecimal representation of " + number + " is: " + hexString);
  }
}
```

Output: Binary representation of 7 is: 111
Octal representation of 7 is: 7

Hexadecimal representation of 7 is: 7

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

```
package Assgn_3;
public class A3_5_l {
        public static void main(String[] args) {
                long longValue = 234557L;
    int intValue = (int) longValue;
    System.out.println("long to int: " + intValue);
    short shortValue = (short) longValue;
    System.out.println("long to short: " + shortValue);
    byte byteValue = (byte) longValue;
    System.out.println("long to byte: " + byteValue);
    float floatValue = (float) longValue;
    System.out.println("long to float: " + floatValue);
    double doubleValue = (double) longValue;
    System.out.println("long to double: " + doubleValue);
    char charValue = (char) (longValue % 65536);
    System.out.println("long to char: " + charValue);
    boolean booleanValue = (longValue != 0);
    System.out.println("long to boolean: " + booleanValue);
    intValue = 2707;
    longValue = intValue;
    System.out.println("int to long: " + longValue);
    shortValue = 2707;
    longValue = shortValue;
    System.out.println("short to long: " + longValue);
    byteValue = 100;
    longValue = byteValue;
    System.out.println("byte to long: " + longValue);
    floatValue = 1298.45f;
    longValue = (long) floatValue;
    System.out.println("float to long: " + longValue);
    doubleValue = 14756.789;
    longValue = (long) doubleValue;
    System.out.println("double to long: " + longValue);
    charValue = 'P';
    longValue = charValue;
    System.out.println("char to long: " + longValue);
    booleanValue = false;
    longValue = booleanValue ? 1L : 0L;
    System.out.println("boolean to long: " + longValue);
                }
}
```

Output: long to int: 234557

long to short: -27587
long to byte: 61
long to float: 234557.0
long to double: 234557.0
long to char: 鏟
long to boolean: true
int to long: 2707
short to long: 2707
byte to long: 100
float to long: 1298
double to long: 14756
char to long: 80
boolean to long: 0

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

**a.** Explore the [Java API documentation for java.lang.Float]() and observe its modifiers and super types.

| Modifier and Type | Field and Description |
|---|---|
| static int | **BYTES**<br>The number of bytes used to represent a `float` value. |
| static int | **MAX_EXPONENT**<br>Maximum exponent a finite `float` variable may have. |
| static float | **MAX_VALUE**<br>A constant holding the largest positive finite value of type `float`, $(2-2^{-23})\cdot 2^{127}$. |
| static int | **MIN_EXPONENT**<br>Minimum exponent a normalized `float` variable may have. |
| static float | **MIN_NORMAL**<br>A constant holding the smallest positive normal value of type `float`, $2^{-126}$. |
| static float | **MIN_VALUE**<br>A constant holding the smallest positive nonzero value of type `float`, $2^{-149}$. |
| static float | **NaN**<br>A constant holding a Not-a-Number (NaN) value of type `float`. |
| static float | **NEGATIVE_INFINITY**<br>A constant holding the negative infinity of type `float`. |
| static float | **POSITIVE_INFINITY**<br>A constant holding the positive infinity of type `float`. |
| static int | **SIZE**<br>The number of bits used to represent a `float` value. |
| static Class<Float> | **TYPE**<br>The `Class` instance representing the primitive type `float`. |

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

A3_6_c.java | A3_6_b.java | A3_6_d.java | A3_6_e.java | A3_6_f.java | A3_6_g.java | A3_6_h.java | A3_6_i.java | A3_6_j.java | A3_6_k.java | A3_6_l.java

```java
package Assgn_3;
public class A3_6_b {
    public static void main(String[] args) {
        int bytesUsed = Float.BYTES;
        System.out.println("The number of bytes used to represent a float value is: " + bytesUsed);
    }
}
```

Console

```
<terminated> A3_6_b [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (07-Sept-2024, 6:49:20 pm – 6:49:20 pm) [pid: 18472]
The number of bytes used to represent a float value is: 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).

```java
package Assgn_3;
public class A3_6_c {
    public static void main(String[] args) {
        float minValue = Float.MIN_VALUE;
        float maxValue = Float.MAX_VALUE;
        System.out.println("The minimum positive non-zero value of float is: " + minValue);
        System.out.println("The maximum value of float is: " + maxValue);
    }
}
```

```
<terminated> A3_6_c [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 6:54:39 pm – 6:54:39 pm) [pid: 7512]
The minimum positive non-zero value of float is: 1.4E-45
The maximum value of float is: 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)).

```java
package Assgn_3;
public class A3_6_m {
    public static void main(String[] args) {
        float number = 456.456f;
        String numberString = Float.toString(number);
        System.out.println("Float to String: " + numberString);
    }
}
```

```
<terminated> A3_6_m [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:00:07 pm – 7:00:08 pm) [pid: 6772]
Float to String: 456.456
```

**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)).

```java
package Assgn_3;
public class A3_6_e {
    public static void main(String[] args) {
        String strNumber = "789.012";
        float parsedNumber = Float.parseFloat(strNumber);
        System.out.println("String to float: " + parsedNumber);
    }
}
```

```
<terminated> A3_6_e [Java Application] C:\Program Files\Java\jdk-21\
String to float: 789.012
```

**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).

```java
package Assgn_3;
public class A3_6_f {
    public static void main(String[] args) {
String strNumber = "Ab12Cd3";
          float malformedNumber = Float.parseFloat(strNumber);
        System.out.println("String to float: " + malformedNumber);
    }
}
```

```
Console
<terminated> A3_6_f [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:06:56 pm – 7:06:56 pm) [pid: 15300]
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 Assgn_3.A3_6_f.main(A3_6_f.java:5)
```

**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)).

```java
package Assgn_3;
public class A3_6_g {
    public static void main(String[] args) {
        float number=2345.796f;
        Float floatWrapper = Float.valueOf(number);
        System.out.println("Float wrapper: " + floatWrapper);
    }
}
```

```
Console
<terminated> A3_6_g [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:10:22 pm – 7:10:22 pm) [pid: 13800]
Float wrapper: 2345.796
```

**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)).

```java
package Assgn_3;
public class A3_6_h {
    public static void main(String[] args) {
        String strNumber = "789.012";
        Float floatWrapper = Float.valueOf(strNumber);
        System.out.println("Float wrapper from String: " + floatWrapper);
    }
}
```

```
Console
<terminated> A3_6_h [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:16:02 pm – 7:16:05 pm) [pid: 11860]
Float wrapper from String: 789.012
```

**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)).

A3_6_h.java | A3_6_i.java | A3_6_j.java | A3_6_k.java | A3_6_l.java | A3_6_m.java

```java
package Assgn_3;
public class A3_6_i {
    public static void main(String[] args) {
        float a = 112.3f;
        float b = 984.5f;
        float sum = Float.sum(a, b);
        System.out.println("Sum of floats: " + sum);
        }
}
```

Console

<terminated> A3_6_i [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:18:40 pm – 7:

```
Sum of floats: 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)).

A3_6_h.java | A3_6_i.java | A3_6_j.java | A3_6_k.java | A3_6_l.java | A3_6_m.java

```java
package Assgn_3;
public class A3_6_j {
    public static void main(String[] args) {
        float a = 109.2f;
        float b = 534.6f;
        float minValue = Float.min(a, b);
        float maxValue = Float.max(a, b);
        System.out.println("Minimum value: " + minValue);
        System.out.println("Maximum value: " + maxValue);
    }
}
```

Console

<terminated> A3_6_j [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:21:52 pm – 7:21:52 pm)

```
Minimum value: 109.2
Maximum value: 534.6
```

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

A3_6_k.java × A3_6_l.java A3_6_m.java

```java
package Assgn_3;
public class A3_6_k {
    public static void main(String[] args) {
        float value = -30.0f;
        double sqrtValue = Math.sqrt(value);
        System.out.println("Square root: " + sqrtValue);
    }
}
```

Console ×

<terminated> A3_6_k [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:24:38 pm – 7:24:38 pm

```
Square root: 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).

A3_6_k.java A3_6_l.java × A3_6_m.java

```java
package Assgn_3;
public class A3_6_l {
    public static void main(String[] args) {
        float a = 0.0f;
        float b = 0.0f;
        float result = a / b;
        System.out.println("Division result: " + result);
    }
}
```

Console ×

<terminated> A3_6_l [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:26:38 pm – 7:26:38

```
Division result: NaN
```

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

A3_6_m.java

```java
package Assgn_3;
public class A3_6_m {
    public static void main(String[] args) {
        float floatValue = 345.456f;
        int intValue = (int) floatValue;
        System.out.println("Float to int: " + intValue);
        short shortValue = (short) floatValue;
        System.out.println("Float to short: " + shortValue);
        byte byteValue = (byte) floatValue;
        System.out.println("Float to byte: " + byteValue);
        double doubleValue = (double) floatValue;
        System.out.println("Float to double: " + doubleValue);
        int intNumber = 100;
        float fromInt = (float) intNumber;
        System.out.println("Int to float: " + fromInt);
        double doubleNumber = 123.456789;
        float fromDouble = (float) doubleNumber;
        System.out.println("Double to float: " + fromDouble);
    }}
```

Console

<terminated> A3_6_m [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (07-Sept-2024, 7:29:56 pm – 7:29:56 pm) [pid

```
Float to int: 345
Float to short: 345
Float to byte: 89
Float to double: 345.45599365234375
Int to float: 100.0
Double to float: 123.45679
```

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. (Hint: Use Double.BYTES).

**c.** Write a program to find the minimum and maximum values of double using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Double.MIN_VALUE and Double.MAX_VALUE).

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

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

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

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

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

**i.** Declare two double variables with values 112.3 and 984.5, and add them using a method from the Double class. (Hint: Use Double.sum(double, double)).

**j.** Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class. (Hint: Use Double.min(double, double) and Double.max(double, double)).

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

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

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

8. Conversion between Primitive Types and Strings

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

First, use the toString method of the corresponding wrapper class. (e.g., Integer.toString()).

Then, use the valueOf method of the String class. (e.g., String.valueOf()).

```
package Assgn_3;
public class A3_8 {
        public static void main(String[] args) {
                int intValue = 56;
    double doubleValue = 4.14159;
    float floatValue = 6.718f;
    boolean booleanValue = true;
    char charValue = 'A';
    long longValue = 123456789L;
    short shortValue = 123;
    byte byteValue = 10;
    String intString1 = Integer.toString(intValue);
    String doubleString1 = Double.toString(doubleValue);
    String floatString1 = Float.toString(floatValue);
    String booleanString1 = Boolean.toString(booleanValue);
    String charString1 = Character.toString(charValue);
    String longString1 = Long.toString(longValue);
    String shortString1 = Short.toString(shortValue);
    String byteString1 = Byte.toString(byteValue);
    String intString2 = String.valueOf(intValue);
    String doubleString2 = String.valueOf(doubleValue);
    String floatString2 = String.valueOf(floatValue);
```

```
    String booleanString2 = String.valueOf(booleanValue);
    String charString2 = String.valueOf(charValue);
    String longString2 = String.valueOf(longValue);
    String shortString2 = String.valueOf(shortValue);
    String byteString2 = String.valueOf(byteValue);
    System.out.println("Primitive to String using toString method:");
    System.out.println("int: " + intString1);
    System.out.println("double: " + doubleString1);
    System.out.println("float: " + floatString1);
    System.out.println("boolean: " + booleanString1);
    System.out.println("char: " + charString1);
    System.out.println("long: " + longString1);
    System.out.println("short: " + shortString1);
    System.out.println("byte: " + byteString1);
    System.out.println("\nPrimitive to String using String.valueOf method:");
    System.out.println("int: " + intString2);
    System.out.println("double: " + doubleString2);
    System.out.println("float: " + floatString2);
    System.out.println("boolean: " + booleanString2);
    System.out.println("char: " + charString2);
    System.out.println("long: " + longString2);
    System.out.println("short: " + shortString2);
    System.out.println("byte: " + byteString2);
        }
}
```

Output: Primitive to String using toString method:
int: 56
double: 4.14159
float: 6.718
boolean: true
char: A
long: 123456789
short: 123
byte: 10

Primitive to String using String.valueOf method:
int: 56
double: 4.14159
float: 6.718
boolean: true
char: A
long: 123456789
short: 123
byte: 10

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).

```
package Assgn_3;
public class A3_9 {
	public static class DefaultValues {
		int intValue;
		double doubleValue;
		float floatValue;
		boolean booleanValue;
		char charValue;
		long longValue;
		short shortValue;
		byte byteValue;
		static int staticIntValue;
		static double staticDoubleValue;
		static float staticFloatValue;
		static boolean staticBooleanValue;
		static char staticCharValue;
		static long staticLongValue;
		static short staticShortValue;
		static byte staticByteValue;
		public static void main (String[] args) {
			DefaultValues defaultValues = new DefaultValues();
		  System.out.println("Instance Variables (Fields) Default Values:");
		  System.out.println("int: " + defaultValues.intValue);
		  System.out.println("double: " + defaultValues.doubleValue);
		  System.out.println("float: " + defaultValues.floatValue);
		  System.out.println("boolean: " + defaultValues.booleanValue);
		  System.out.println("char: '" + defaultValues.charValue + "'");
		  System.out.println("long: " + defaultValues.longValue);
		  System.out.println("short: " + defaultValues.shortValue);
		  System.out.println("byte: " + defaultValues.byteValue);
		  System.out.println("\nStatic Variables Default Values:");
		  System.out.println("int: " + staticIntValue);
		  System.out.println("double: " + staticDoubleValue);
		  System.out.println("float: " + staticFloatValue);
		  System.out.println("boolean: " + staticBooleanValue);
		  System.out.println("char: '" + staticCharValue + "'");
		  System.out.println("long: " + staticLongValue);
		  System.out.println("short: " + staticShortValue);
		  System.out.println("byte: " + staticByteValue);
		}
	      }
}
```

```
Instance Variables (Fields) Default Values:
int: 0
double: 0.0
```

float: 0.0
boolean: false
char: '28 ‘
long: 0
short: 0
byte:0
Static Variables Default Values:
int: 0
double: 0.0
float: 0.0
boolean: false
char: '28 ‘
long: 0
short: 0
byte: 0

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).

```
package Assgn_3;

public class A3_10 {
                    public static void main(String[] args) {
                      if (args.length != 3) {
                  System.out.println("Enter ArithmeticOperations: ");
                  System.exit(1);
                }
                int n1;
                int n2;
                n1 = Integer.parseInt(args[0]);
                n2 = Integer.parseInt(args[1]);
                String operator = args[2];
                double result = 0.0;
                boolean vOperator = true;
                switch (operator) {
                  case "+":
                    result = n1 + n2;
                    break;
                  case "-":
                    result = n1 - n2;
                    break;
                  case "*":
                    result = n1 * n2;
                    break;
                  case "/":
                    if (n2 == 0) {
                      System.out.println("Error: Division by zero is not allowed.");
```

```
                vOperator = false;
            } else {
                result = (double) n1 / n2;
            }
            break;
            default:
            System.out.println("Error: Invalid operator. Use +, -, *, or /.");
            vOperator = false;
            break;
        }
        if(vOperator) {
System.out.printf("Result: %.2f%n", result);
        }


            }
}
```