ASSIGNMENT NO.2

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.

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)).

Program:

```
public class bool {

   public static void main(String[] args) {

      boolean status = true ;

      String stringstr = Boolean.toString(status);

      System.out.println(stringstr);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice>java bool.java
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)).

Program:

```
public class pboolean{

public static void main(String args[]){

        String strStatus =new String("true");
```

```
        boolean boolstatus=Boolean.parseBoolean(strStatus);

        System.out.println(" String to boolean  "+boolstatus);

        }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice>java bool1.java
 String to boolean  true
```

d. Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean. (Hint: parseBoolean method will not work as expected with "1" or "0").

Program:

```
public class BooleanConversion {

   public static void main(String[] args) {

      String strStatus = "1";  // or "0"

      boolean boolStatus = Boolean.parseBoolean(strStatus);

      System.out.println("Converted boolean value: " + boolStatus);

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice>java bool2.java
Converted boolean value: false
```

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)).

Program:

```
public class BooleanConversion{

   public static void main(String[] args) {

      boolean status = true;

      Boolean wrappedStatus = Boolean.valueOf(status);  // Autoboxing also
works

      System.out.println("Wrapped boolean value: " + wrappedStatus);

   }
```

```
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice>java bool3.java
Wrapped boolean value: 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)).

Program:

```
public class BooleanConversion {

   public static void main(String[] args) {

      String strStatus = "true";

      Boolean wrappedStatus = Boolean.valueOf(strStatus);

      System.out.println(wrappedStatus);

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice>java bool4.java
true
```

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

Program:

```
public class Qg{

public static void main(String args [])

{

boolean status=false;

int strbool = (status) ? 1 :0;

System.out.println("Boolean to Int: "+strbool);

}

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Boolean>java bool5.java
Boolean to Int: 0
```

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. (Hint: Use Byte.BYTES).

program:

```
class Byteb{

        public static void main(String args[]){

                int bytesUsed = Byte.BYTES;

                System.out.println("Bytes used to represent a byte value: "
+bytesUsed);

        }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java Byteb.java
Bytes used to represent a byte value: 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).

Program:

```
        public class ByteExample {

          public static void main(String[] args) {

              System.out.println("Minimum byte value: " + Byte.MIN_VALUE);

              System.out.println("Maximum byte value: " + Byte.MAX_VALUE);

            }

        }

        }
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java Bytec.java
Minimum byte value: -128
Maximum byte value: 127
```

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)).

Program:

```
public class ByteExample {

   public static void main(String[] args) {

      byte number = 100;

      String strNumber = Byte.toString(number);

      System.out.println(strNumber);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java ByteD.java
100
```

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)).

Program:

```
public class ByteE{

   public static void main(String[] args) {

      byte number = 100;

      String strNumber = Byte.toString(number);

      System.out.println(strNumber);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java ByteE.java
100
```

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).

Program:

```
public class ByteF{

   public static void main(String[] args) {

      String strNumber = "Ab12cd3";

      byte number = Byte.parseByte(strNumber);

      System.out.println(number);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java Bytef.java
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 ByteF.main(Bytef.java:5)
```

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)).

Program:

```
public class ByteG{

   public static void main(String[] args) {

      byte number = 42;

      Byte byteObject = Byte.valueOf(number);

      System.out.println(byteObject);
```

```
    }
  }
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java ByteG.java
42
```

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)).

Program:

```
public class ByteH {
  public static void main(String[] args) {
    String strNumber = "127";
    Byte byteObject = Byte.valueOf(strNumber);
    System.out.println("Byte object from string: " + byteObject);
  }
}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte> java ByteH.java
127
```

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

```
public class ByteI{
  public static void main(String[] args) {
    byte number = 10;
    int intValue = number;
    short shortValue = number;
    long longValue = number;
    float floatValue = number;
```

```
        double doubleValue = number;


        System.out.println("Byte value as int: " + intValue);

        System.out.println("Byte value as short: " + shortValue);

        System.out.println("Byte value as long: " + longValue);

        System.out.println("Byte value as float: " + floatValue);

        System.out.println("Byte value as double: " + doubleValue);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Byte>java ByteI.java
Byte value as int: 10
Byte value as short: 10
Byte value as long: 10
Byte value as float: 10.0
Byte value as double: 10.0
```

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. (Hint: Use Short.BYTES).

program:

```
public class ShortB {

    public static void main(String[] args) {

        System.out.println("Bytes used to represent a short value: " +
Short.BYTES);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortB.java
Bytes used to represent a short value: 2
```

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

Program:

```
public class ShortC{

   public static void main(String[] args) {

      System.out.println("Minimum short value: " + Short.MIN_VALUE);

      System.out.println("Maximum short value: " + Short.MAX_VALUE);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortC.java
Minimum short value: -32768
Maximum short value: 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)).

Program:

```
public class ShortD{

   public static void main(String[] args) {

      short number = 32000;

      String strNumber = Short.toString(number);

      System.out.println("String representation of short value: " + strNumber);

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortD.java
String representation of short value: 32000
```

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)).

Program:

```
public class ShortE {

   public static void main(String[] args) {

      String strNumber = "12345";

      short number = Short.parseShort(strNumber);

      System.out.println("Short value from string: " + number);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortE.java
Short value from string: 12345
```

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).

Program:

```
public class ShortF {

   public static void main(String[] args) {

      String strNumber = "Ab12cd3";

      short number = Short.parseShort(strNumber);

      System.out.println("Short value from string: " + number);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortF.java
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 ShortF.main(shortF.java:5)
```

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)).

Program:

```
public class ShortG{

   public static void main(String[] args) {

      short number = 10000;

      Short wrapper = Short.valueOf(number);

      System.out.println("Short object: " + wrapper);

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortG.java
Short object: 10000
```

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)).

Program:

```
public class ShortH{

   public static void main(String[] args) {

      String strNumber = "32767";

      Short shortval = Short.valueOf(strNumber);

      System.out.println("Short object from string: " + shortval);

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortH.java
Short object from string: 32767
```

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

Program:

```
public class ShortI {
   public static void main(String[] args) {
```

```
        short number = 100;
        int intValue = number;
        byte byteValue = (byte) number; // Casting needed
        long longValue = number;
        float floatValue = number;
        double doubleValue = number;

        System.out.println("Short value as int: " + intValue);
        System.out.println("Short value as byte (with casting): " +
byteValue);
        System.out.println("Short value as long: " + longValue);
        System.out.println("Short value as float: " + floatValue);
        System.out.println("Short value as double: " + doubleValue);
    }
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Short>java shortI.java
Short value as int: 100
Short value as byte (with casting): 100
Short value as long: 100
Short value as float: 100.0
Short value as double: 100.0
```

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. (Hint: Use Integer.BYTES).

program:

```
public class IntegerExample {

   public static void main(String[] args) {

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

   }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java Integerb.java
Bytes used to represent an int value: 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).

Program:

```
public class IntegercC {

   public static void main(String[] args) {

      System.out.println("Minimum int value: " + Integer.MIN_VALUE);

      System.out.println("Maximum int value: " + Integer.MAX_VALUE);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegercC.java
Minimum int value: -2147483648
Maximum int value: 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)).

Program:

```
public class IntegerExample {

   public static void main(String[] args) {

      int number = 100;

      String strNumber = Integer.toString(number);

      System.out.println("String representation of int value: " + strNumber);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerD.java
String representation of int value: 100
```

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)).

Program:

```
public class IntegerExample {

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

```
        String strNumber = "123";

        int number = Integer.parseInt(strNumber);

        System.out.println("Int value from string: " + number);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerE.java
Int value from string: 123
```

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

Program:

```
public class ShortF {

    public static void main(String[] args) {

        String strNumber = "Ab12cd3";

        int number = Integer.parseInt(strNumber);

        System.out.println("int value from string: " + number);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerF.java
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 ShortF.main(IntegerF.java:5)
```

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)).

Program:

```
public class IntegerG{

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

```
        int number = 100;

        Integer wrapperObject = Integer.valueOf(number);

        System.out.println("Integer object: " + wrapperObject);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerG.java
Integer object: 100
```

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)).

Program:

```
public class IntegerExample {

    public static void main(String[] args) {

        String strNumber = "456";

        int wrapperObject = Integer.valueOf(strNumber);

        System.out.println("Integer object from string: " + wrapperObject);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerH.java
Integer object from string: 456
```

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)).

Program:

```
public class IntegerExample {

    public static void main(String[] args) {

        int a = 10;

        int b = 20;
```

```
        int sum = Integer.sum(a, b);

        System.out.println("Sum of 10 and 20: " + sum);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerI.java
Sum of 10 and 20: 30
```

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

Program:

```
public class IntegerJ{

    public static void main(String[] args) {

        int a = 10;

        int b = 20;

        int minValue = Integer.min(a, b);

        int maxValue = Integer.max(a, b);

        System.out.println("Minimum value between 10 and 20: " + minValue);

        System.out.println("Maximum value between 10 and 20: " + maxValue);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerJ.java
Minimum value between 10 and 20: 10
Maximum value between 10 and 20: 20
```

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

Program:

```
public class IntegerExample {
```

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

        int number = 7;

        String binary = Integer.toBinaryString(number);

        String octal = Integer.toOctalString(number);

        String hex = Integer.toHexString(number);

        System.out.println("Binary representation of 7: " + binary);

        System.out.println("Octal representation of 7: " + octal);

        System.out.println("Hexadecimal representation of 7: " + hex);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerK.java
Binary representation of 7: 111
Octal representation of 7: 7
Hexadecimal representation of 7: 7

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>
```

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

Program:

```
public class IntegerExample {

    public static void main(String[] args) {

        int number = 45 ;

        byte byteValue = (byte) number;  // Explicit casting needed

        short shortValue = (short) number;  // Explicit casting

        long longValue = number;

        float floatValue = number;

        double doubleValue = number;


        System.out.println("Int value as byte : " + byteValue); // with casting
```

```
        System.out.println("Int value as short : " + shortValue);   //with casting

        System.out.println("Int value as long: " + longValue);

        System.out.println("Int value as float: " + floatValue);

        System.out.println("Int value as double: " + doubleValue);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\Integer>java IntegerL.java
Int value as byte : 45
Int value as short : 45
Int value as long: 45
Int value as float: 45.0
Int value as double: 45.0
```

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. (Hint: Use Long.BYTES).

program:

```
public class LongBytes {

    public static void main(String[] args) {

        System.out.println("Bytes used by long: " + Long.BYTES);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long1.java
Bytes used by long: 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).

Program:

```
public class LongMinMax {

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

```
        System.out.println("Min value of long: " + Long.MIN_VALUE);

        System.out.println("Max value of long: " + Long.MAX_VALUE);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long2.java
Min value of long: -9223372036854775808
Max value of long: 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)).

Program:

```
public class LongToString {

    public static void main(String[] args) {

        long number = 12345L; // Method-local variable

        String strNumber = Long.toString(number); // Conversion to String

        System.out.println("String representation of long: " + strNumber);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long3.java
String representation of long: 12345
```

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)).

Program:

```
public class StringToLong {

    public static void main(String[] args) {

        String strNumber = "54321"; // Method-local variable

        long number = Long.parseLong(strNumber); // Conversion to long
```

```
        System.out.println("Converted long value: " + number);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long4.java
Converted long value: 54321
```

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).

Program:

```
class InvalidStringToLong {

    public static void main(String[] args) {

        String strNumber = "Ab12Cd3"; // Method-local variable

         System.out.println("NumberFormatException: " + e.getMessage());



    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long5.java
long5.java:5: error: cannot find symbol
         System.out.println("NumberFormatException: " + e.getMessage());
                                                        ^
  symbol:   variable e
  location: class InvalidStringToLong
1 error
error: compilation failed
```

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)).

Program:

```
public class LongToWrapper {

    public static void main(String[] args) {

        long number = 98765L; // Method-local variable

        Long longWrapper = Long.valueOf(number); // Conversion to Long
wrapper class
```

```
        System.out.println("Long wrapper object: " + longWrapper);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long6.java
Long wrapper object: 98765
```

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)).

Program:

```
public class StringToWrapper {

    public static void main(String[] args) {

        String strNumber = "67890"; // Method-local variable

        Long longWrapper = Long.valueOf(strNumber); // Conversion to Long
wrapper class

        System.out.println("Long wrapper object: " + longWrapper);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long7.java
Long wrapper object: 67890
```

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)).

Program:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\JAVA\Day 3\Day 3 practice\long>java long8.java
Sum of 1123 and 9845 is: 10968
```

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

Program:

```
public class long9{
    public static void main(String[] args) {
        long num1 = 1122;
        long num2 = 5566;
        long min = Long.min(num1, num2);
        long max = Long.max(num1, num2);
        System.out.println("Minimum value: " + min);
        System.out.println("Maximum value: " + max);
    }
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\long>java l
ong9.java
Minimum value: 1122
Maximum value: 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)).

Program:

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

```
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\long>java l
ong10.java
Binary representation: 111
Octal representation: 7
Hexadecimal representation: 7
```

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

Program code:

```
public class LongConversionExperiment {
    public static void main(String[] args) {
        long num = 123456789L;
        int intValue = (int) num;
        short shortValue = (short) num;
        byte byteValue = (byte) num;
        double doubleValue = (double) num;
        float floatValue = (float) num;

        System.out.println("Original long value: " + num);
        System.out.println("Converted to int: " + intValue);
        System.out.println("Converted to short: " + shortValue);
        System.out.println("Converted to byte: " + byteValue);
        System.out.println("Converted to double: " + doubleValue);
        System.out.println("Converted to float: " + floatValue);
    }
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\long>java L
ong11.java
Original long value: 123456789
Converted to int: 123456789
Converted to short: -13035
Converted to byte: 21
Converted to double: 1.23456789E8
Converted to float: 1.23456792E8
```

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. (Hint: Use Float.BYTES).

program:

```
public class FloatB {

   public static void main(String[] args) {

      System.out.println("Size: " + Float.BYTES);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatB.java
Size: 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).

Program:

```
public class FloatC{

   public static void main(String[] args) {

      System.out.println("Minimum float value: " + Float.MIN_VALUE);

      System.out.println("Maximum float value: " + Float.MAX_VALUE);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2
floatC.java
Minimum float value: 1.4E-45
Maximum float value: 3.4028235E38
```

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

Program:

```
public class floatD {

   public static void main(String[] args) {

      float number = 123.45f;

      String str = Float.toString(number);

      System.out.println("Float to String: " + str);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Modul
floatD.java
Float to String: 123.45
```

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)).

Program:

```
public class floatE {

   public static void main(String[] args) {

      String strNumber = "123.45";

      float number = Float.parseFloat(strNumber);

      System.out.println("String to float: " + number);
```

```
    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatE.java
String to float: 123.45
```

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).

Program:

```
public class floatF {

    public static void main(String[] args) {

        String strNumber = "Ab12Cd3";

        float number = Float.parseFloat(strNumber);

        System.out.println("Converted value: " + number);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatF.java
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 floatF.main(floatF.java:7)
```

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

Program:

```
public class floatG {

    public static void main(String[] args) {

        float a = 112.2f;

        float b = 556.6f;

        float min = Float.min(a, b);
```

```
            float max = Float.max(a, b);

            System.out.println("Minimum value: " + min);

            System.out.println("Maximum value: " + max);

        }

    }
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatG.java
Minimum value: 112.2
Maximum value: 556.6
```

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)).

Program:

```
public class floatH {

  public static void main(String[] args) {

    String strNumber = "123.45";

    Float floatWrapper = Float.valueOf(strNumber);

    System.out.println("String to Float wrapper: " + floatWrapper);

  }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatH.java
String to Float wrapper: 123.45

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>
```

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)).

Program:

```
        public class FloatAddition {

            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 112.3 and 984.5: " + sum);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatI.java
Sum of 112.3 and 984.5: 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)).

Program:

```
public class floatJ {

    public static void main(String[] args) {

        float a = 112.2f;

        float b = 556.6f;

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

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

        System.out.println("Minimum value: " + min);

        System.out.println("Maximum value: " + max);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatJ.java
Minimum value: 112.2
Maximum value: 556.6
```

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

Program:

```
public class floatK {

   public static void main(String[] args) {

      float number = -25.0f;

      double sqrt = Math.sqrt(number);

      System.out.println("Square root of -25.0: " + sqrt);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatK.java
Square root of -25.0: 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).

Program:

```
public class floatL {

   public static void main(String[] args) {

      float a = 0.0f;

      float b = 0.0f;

      float result = a / b;

      System.out.println("0.0 / 0.0 = " + result);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatL.java
0.0 / 0.0 = NaN
```

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

Program:

```
public class floatM {

  public static void main(String[] args) {

    float floatValue = 123.45f;

    int intValue = (int) floatValue;

    System.out.println("Float to int: " + intValue);

    double doubleValue = floatValue;

    System.out.println("Float to double: " + doubleValue);

    intValue = 123;

    float floatFromInt = (float) intValue;

    System.out.println("Int to float: " + floatFromInt);

    doubleValue = 123.45;

    float floatFromDouble = (float) doubleValue;

    System.out.println("Double to float: " + floatFromDouble);

  }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Float>java
floatM.java
Float to int: 123
Float to double: 123.44999694824219
Int to float: 123.0
Double to float: 123.45
```

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).

program:

```
public class DoubleB {

   public static void main(String[] args) {

      System.out.println("Number of bytes used to represent a double: " +
Double.BYTES);

   }

}
```

output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleB.java
Number of bytes used to represent a double: 8
```

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

Program:

```
public class doubleC {

   public static void main(String[] args) {

      System.out.println("Minimum value of double: " + Double.MIN_VALUE);

      System.out.println("Maximum value of double: " + Double.MAX_VALUE);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleC.java
Minimum value of double: 4.9E-324
Maximum value of double: 1.7976931348623157E308

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>
```

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)).

Program:

```
public class doubleD {

   public static void main(String[] args) {

      double number = 123.456;

      String strNumber = Double.toString(number);

      System.out.println("String representation of the double value: " +
strNumber);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleD.java
String representation of the double value: 123.456
```

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)).

Program:

```
public class  doubeE{

   public static void main(String[] args) {

      String strNumber = "456.789";

      double number = Double.parseDouble(strNumber);

      System.out.println("Double representation of the string value: " +
number);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleE.java
Double representation of the string value: 456.789
```

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).

Program:

```
public class doubleF {

   public static void main(String[] args) {

      String strNumber = "Ab12Cd3"; // Declare method-local variable with the given value



      float number = Float.parseFloat(strNumber);

      System.out.println("Converted value: " + number);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\
 doubleF.java
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 doubleF.main(doubleF.java:7)
```

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)).

Program:

```
public class DoubleValueOfTest {

   public static void main(String[] args) {

      double number = 789.123;

      Double wrapperDouble = Double.valueOf(number);

      System.out.println("Double wrapper class object: " + wrapperDouble);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA
 doubleG.java
Double wrapper class object: 789.123
```

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)).

Program:

```
public class StringToDoubleValueOfTest {

   public static void main(String[] args) {

      String strNumber = "123.456";

      Double WDS= Double.valueOf(strNumber);

      System.out.println("Double wrapper class object: " +WDS);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleH.java
Double wrapper class object: 123.456
```

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)).

Program:

```
public class doubleI {

   public static void main(String[] args) {

      double num1 = 112.3;

      double num2 = 984.5;

      double sum = Double.sum(num1, num2);

      System.out.println("Sum of the two double values: " + sum);

   }
```

```
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleI.java
Sum of the two double values: 1096.8
```

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)).

Program:

```
public class DoubleMinMaxTest {

   public static void main(String[] args) {

      double num1 = 112.2;

      double num2 = 556.6;

      double min = Double.min(num1, num2);

      double max = Double.max(num1, num2);

      System.out.println("Minimum value: " + min);

      System.out.println("Maximum value: " + max);

   }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignments\Sandeep sir assignments\Assignment 2\ass2\Double>java
 doubleJ.java
Minimum value: 112.2
Maximum value: 556.6
```

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

Program:

```
public class doubleK {

   public static void main(String[] args) {

      double number = -25.0;

      double sqrt = Math.sqrt(number);
```

```
        System.out.println("Square root of " + number + " is: " + sqrt);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JA
 doubleK.java
Square root of -25.0 is: NaN

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JA
```

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

Program::

```
public class doubleL {

    public static void main(String[] args) {

        double num1 = 0.0;

        double num2 = 0.0;

        double result = num1 / num2;

        System.out.println("Result of dividing 0.0 by 0.0: " + result);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA
 doubleL.java
Result of dividing 0.0 by 0.0: NaN
```

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

Program:

```
public class doubleM {

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

```
        double number = 123.456;

        int intValue = (int) number;

        float floatValue = (float) number;

        long longValue = (long) number;

        System.out.println("Double value: " + number);

        System.out.println("Converted to int: " + intValue);

        System.out.println("Converted to float: " + floatValue);

        System.out.println("Converted to long: " + longValue);

    }

}
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC
 doubleM.java
Double value: 123.456
Converted to int: 123
Converted to float: 123.456
Converted to long: 123
```

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()).

Program:

```
public class Primitive {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter an integer: ");

        int intVal = sc.nextInt();

        System.out.print("Enter a double: ");
```

```
    double doubleVal = sc.nextDouble();

    System.out.print("Enter a boolean: ");

    boolean boolVal = sc.nextBoolean();

    // Converting using toString() method of wrapper classes

    String intToString = Integer.toString(intVal);

    String doubleToString = Double.toString(doubleVal);

    String boolToString = Boolean.toString(boolVal);

    System.out.println("\nConversion using toString() method:");

    System.out.println("Integer as String: " + intToString);

    System.out.println("Double as String: " + doubleToString);

    System.out.println("Boolean as String: " + boolToString);
```

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

```
public class Primitive {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);


    System.out.print("Enter an integer: ");

    int intVal = sc.nextInt();

    System.out.print("Enter a double: ");

    double doubleVal = sc.nextDouble();

    System.out.print("Enter a boolean: ");

    boolean boolVal = sc.nextBoolean();

 // Converting using valueOf() method of String class

    String intValueOf = String.valueOf(intVal);

    String doubleValueOf = String.valueOf(doubleVal);
```

```
String boolValueOf = String.valueOf(boolVal);


System.out.println("\nConversion using valueOf() method:");

System.out.println("Integer as String: " + intValueOf);

System.out.println("Double as String: " + doubleValueOf);

System.out.println("Boolean as String: " + boolValueOf);
```

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\Assignme
java primitive.jav
Enter an integer: 15
Enter a double: 45.00
Enter a boolean: true

Conversion using toString() method:
Integer as String: 15
Double as String: 45.0
Boolean as String: true

Conversion using valueOf() method:
Integer as String: 15
Double as String: 45.0
Boolean as String: true
```

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).

Program:

```
public class DefaultValues {

  // Instance variables (have default values)

  byte byteVal;

  short shortVal;

  int intVal;

  long longVal;
```

```
float floatVal;
double doubleVal;
char charVal;
boolean booleanVal;

// Static variables (also have default values)
static byte staticByteVal;
static short staticShortVal;
static int staticIntVal;
static long staticLongVal;
static float staticFloatVal;
static double staticDoubleVal;
static char staticCharVal;
static boolean staticBooleanVal;

public static void main(String[] args) {
    // Create an instance to check instance variable default values
    DefaultValues obj = new DefaultValues();

    // Display default values of instance variables
    System.out.println("Default values of instance variables:");
    System.out.println("byte: " + obj.byteVal);
    System.out.println("short: " + obj.shortVal);
    System.out.println("int: " + obj.intVal);
    System.out.println("long: " + obj.longVal);
    System.out.println("float: " + obj.floatVal);
```

```
        System.out.println("double: " + obj.doubleVal);

        System.out.println("char: [" + obj.charVal + "]"); // Displays an empty space

        System.out.println("boolean: " + obj.booleanVal);



        // Display default values of static variables

        System.out.println("\nDefault values of static variables:");

        System.out.println("byte: " + staticByteVal);

        System.out.println("short: " + staticShortVal);

        System.out.println("int: " + staticIntVal);

        System.out.println("long: " + staticLongVal);

        System.out.println("float: " + staticFloatVal);

        System.out.println("double: " + staticDoubleVal);

        System.out.println("char: [" + staticCharVal + "]");

        System.out.println("boolean: " + staticBooleanVal);

    }

}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAV
java DefaultValues.jav
Default values of instance variables:
byte: 0
short: 0
int: 0
long: 0
float: 0.0
double: 0.0
char: []
boolean: false

Default values of static variables:
byte: 0
short: 0
int: 0
long: 0
float: 0.0
double: 0.0
char: []
boolean: false
```

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).

Program:

```
public class ArithmeticOperations {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);



        // Input two integers

        System.out.print("Enter the first integer: ");

        int num1 = scanner.nextInt();

        System.out.print("Enter the second integer: ");

        int num2 = scanner.nextInt();



        // Input the operator

        System.out.print("Enter an arithmetic operator (+, -, *, /): ");

        char operator = scanner.next().charAt(0);
```

```
        // Perform the operation using switch-case
        switch (operator) {
            case '+':
                System.out.println("Result: " + (num1 + num2));
                break;
            case '-':
                System.out.println("Result: " + (num1 - num2));
                break;
            case '*':
                System.out.println("Result: " + (num1 * num2));
                break;
            case '/':
                if (num2 != 0) {
                    System.out.println("Result: " + (num1 / num2));
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Error: Invalid operator.");
        }

        scanner.close();
    }
}
```

Output:

```
C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\A
>java B10.java
Enter the first integer: 10
Enter the second integer: 15
Enter an arithmetic operator (+, -, *, /): +
Result: 25

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\A
>java B10.java
Enter the first integer: 25
Enter the second integer: 15
Enter an arithmetic operator (+, -, *, /): -
Result: 10

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\A
>java B10.java
Enter the first integer: 45
Enter the second integer: 18
Enter an arithmetic operator (+, -, *, /): *
Result: 810

C:\Users\rahul\OneDrive\Desktop\CDAC\Module 2 JAVA OOP\A
>java B10.java
Enter the first integer: 55
Enter the second integer: 5
Enter an arithmetic operator (+, -, *, /): /
Result: 11
```