

Note: Consider the following before starting the assignment:

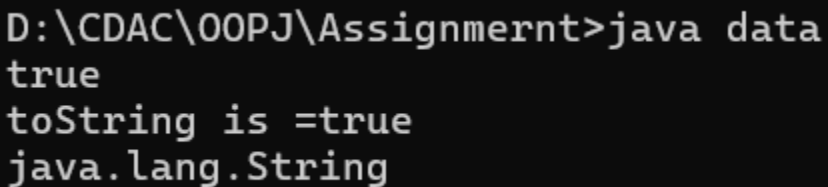
- A static field declared inside a class is called a class-level variable. To access this variable, use the class name and the dot operator (e.g., `Integer.MAX_VALUE`).
- A static method defined inside a class is called a class-level method. To access this method, use the class name and the dot operator (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`

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

```
class data{
    public static void main(String args[]){
        boolean status = true;
        System.out.println("true");
        //System.out.println((Object(status)).getClass().getName());

        String sString = Boolean.toString(status);
        System.out.println("toString is =" + sString);
        System.out.println(sString.getClass().getName());
    }
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data
true
toString is =true
java.lang.String
```

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)`).

```
class data1{

    public static void main(String args[]){

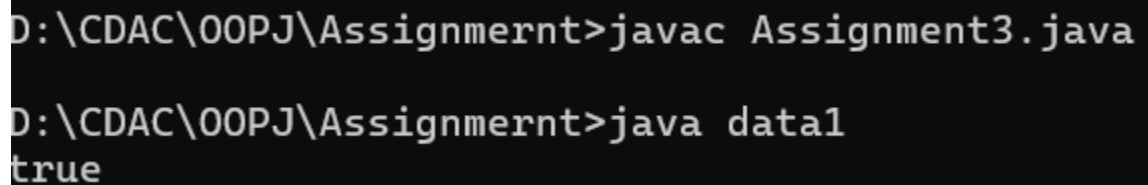
        String strStatus = "true";

        boolean pBoolean = Boolean.parseBoolean(strStatus);

        System.out.println(pBoolean);

    }

}
```



```
D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java
D:\CDAC\OOPJ\Assignmernt>java data1
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"`).

```
class data2{

    public static void main(String args[]){

        String strStatus = "0";

        boolean pBoolean = "1".equals(strStatus);

        System.out.println(pBoolean);

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data2
false
```

```
D:\CDAC\00PJ\Assignmernt>java data2
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)`).

```
class data3{

    public static void main(String args[]){

        Boolean a;

        boolean status = true;

        a = Boolean.valueOf(status);

        System.out.println(a);

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data3
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)`).

```
class data4{

    public static void main(String args[]){

        String strStatus = "true";

        Boolean a;
```

```

        a = Boolean.valueOf(strStatus);

        System.out.println(a);
    }
}

```

```

D:\CDAC\OOPJ\Assignmernt>java data4
true

```

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

```

class data5{

    public static void main(String args[]){

        boolean strStatus = true;

        int number = strStatus?1:0;

        System.out.println(number);

    }

}

class data6{

    public static void main(String args[]){

        char charStatus = 'F';

        boolean status = (charStatus == 'T');

        System.out.println(status);

    }

}

```

```
D:\CDAC\OOPJ\Assignmernt>java data5
1
```

```
D:\CDAC\OOPJ\Assignmernt>java data6
false
```

2. Working with `java.lang.Byte`

- Explore the [Java API documentation for `java.lang.Byte`](#) and observe its modifiers and super types.

```
class data7{

    public static void main(String args[]){

        byte value = Byte.BYTES;

        System.out.println(value);

        long value1 = Long.BYTES;

        System.out.println(value1);

        double value2 = Character.BYTES;

        System.out.println(value2);

    }

}
```

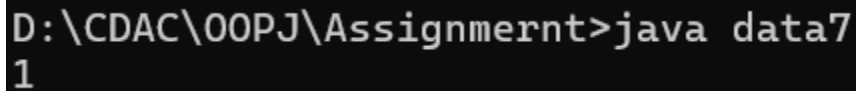
```
D:\CDAC\OOPJ\Assignmernt>java data7
1
8

D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java

D:\CDAC\OOPJ\Assignmernt>java data7
1
8
2.0
```

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`).

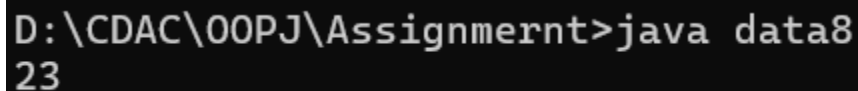
```
class data7{  
  
    public static void main(String args[]){  
  
        byte value = Byte.BYTES;  
  
        System.out.println(value);  
  
    }  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data7  
1
```

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)`).

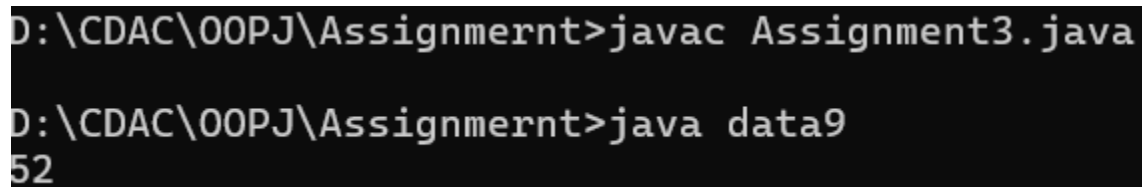
```
class data8{  
  
    public static void main(String args[]){  
  
        byte byStatus = 23;  
  
        String a = Byte.toString(byStatus);  
  
        System.out.println(a);  
  
    }  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data8  
23
```

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)`).

```
class data9{  
    public static void main(String args[]){  
        String strNumber = "52";  
        byte a = Byte.parseByte(strNumber);  
        System.out.println(a);  
    }  
}
```



The screenshot shows a terminal window with a black background and white text. The first line shows the command `D:\CDAC\00PJ\Assignmernt>javac Assignment3.java`. The second line shows the command `D:\CDAC\00PJ\Assignmernt>java data9`. The output of the program is `52`.

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`).

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)`).

```
class data11{  
    public static void main(String args[]){  
        byte byvalue = 2;  
  
        System.out.println(Byte.valueOf(byvalue));  
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data11
2
```

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)`).

```
class data12{

    public static void main(String args[]){

        String strNumber = "20";

        System.out.println(Byte.valueOf(strNumber));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java

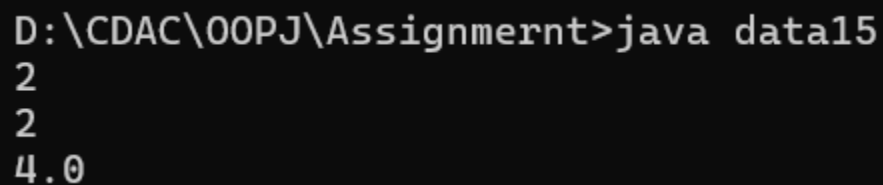
D:\CDAC\OOPJ\Assignmernt>java data12
20
```

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

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`).

```
class data15{  
  
    public static void main(String args[]){  
  
        int Number = Short.BYTES;  
  
        System.out.println(Number);  
  
        long Number1 = Short.BYTES;  
  
        System.out.println(Number1);  
  
        double Number2 = Float.BYTES;  
  
        System.out.println(Number2);  
  
    }  
  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data15  
2  
2  
4.0
```

- 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`).

```
class data16{  
  
    public static void main(String args[]){
```

```

        int Number = Short.MIN_VALUE;

        System.out.println(Number);

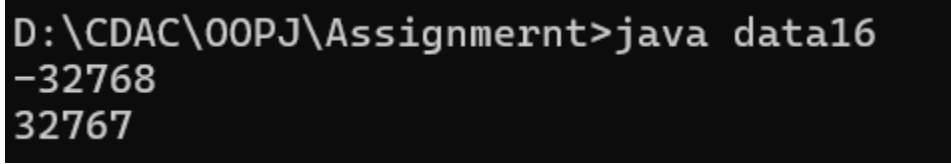
        int Number1 = Short.MAX_VALUE;

        System.out.println(Number1);

    }

}

```



```

D:\CDAC\OOPJ\Assignmernt>java data16
-32768
32767

```

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)`).

```

class data17{

    public static void main(String args[]){

        int Number = 45;

        System.out.println(Integer.toString(Number));

        void printType(Number) {

            System.out.print("Int");

        }

    }

}

```

```
D:\CDAC\00PJ\Assignmernt>java data17
45
```

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)`)

```
class data18{

    public static void main(String args[]){

        String strNumber = "45";

        System.out.println(Byte.parseByte(strNumber));

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data18
45
```

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`).

```
D:\CDAC\00PJ\Assignmernt>java data19
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 data19.main(Assignment3.java:223)
```

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)`).

```
class data20{  
  
    public static void main(String args[]){  
  
        byte number = 3;  
  
        System.out.println(Byte.valueOf(number));  
  
    }  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data20  
3
```

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)`).

```
class data21{  
  
    public static void main(String args[]){  
  
        String strNumber = "3";  
  
        System.out.println(Byte.valueOf(strNumber));  
  
    }  
}
```

```
D:\CDAC\00PJ\Assignmernt>javac Assignment3.java
D:\CDAC\00PJ\Assignmernt>java data21
3
```

i. 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)`).

```
class data22{

    public static void main(String args[]){

        String strNumber = "2";

        System.out.println(Short.valueOf(strNumber));

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data22
2
```

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`).

```
class data24{

    public static void main(String args[]){

        int value = Integer.BYTES;

        System.out.println(value);

    }

}
```

```
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data24  
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`).

```
class data25{  
  
    public static void main(String args[]){  
  
        int value = Integer.MIN_VALUE;  
  
        System.out.println(value);  
  
        int value1 = Integer.MAX_VALUE;  
  
        System.out.println(value1);  
  
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data25  
-2147483648  
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)`).

```
class data26{

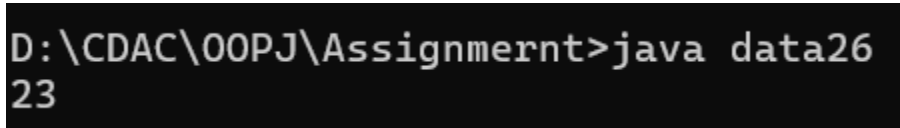
    public static void main(String args[]){

        int number = 23;

        System.out.println(Integer.toString(number));

    }

}
```



```
D:\CDAC\00PJ\Assignmernt>java data26
23
```

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)`).

```
class data27{

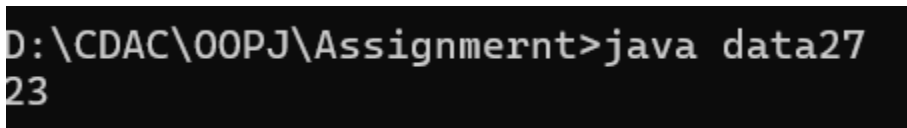
    public static void main(String args[]){

        String strNumber = "23";

        System.out.println(Integer.parseInt(strNumber));

    }

}
```



```
D:\CDAC\00PJ\Assignmernt>java data27
23
```

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`).

```
class data28{

    public static void main(String args[]){

        String strNumber = "Ab12Cd3";

        System.out.println(Integer.parseInt(strNumber));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data28
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 data28.main(Assignment3.java:329)
```

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)`).

```
class data29{

    public static void main(String args[]){

        int number = 50;

        System.out.println(Integer.valueOf(number));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java

D:\CDAC\OOPJ\Assignmernt>java data29
50
```


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)`).

```
class data30{

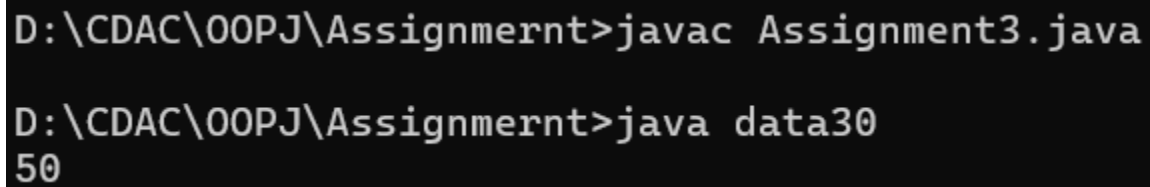
    public static void main(String args[]){

        String strNumber = "50";

        System.out.println(Integer.valueOf(strNumber));

    }

}
```



```
D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java
D:\CDAC\OOPJ\Assignmernt>java data30
50
```

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)`).

```
class data31{

    public static void main(String args[]){

        int num1 = 10;

        int num2 = 20;

        System.out.println(Integer.sum(num1, num2));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>javac Assignment3.java
D:\CDAC\OOPJ\Assignmernt>java data31
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)`).

```
class data32{

    public static void main(String args[]){

        int num1 = 10;

        int num2 = 20;

        System.out.println(Integer.min(num1, num2));

        System.out.println(Integer.max(num1, num2));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data32
10
20
```

```
class data33{

    public static void main(String args[]){

        int num1 = 7;
```

```

        System.out.println(Integer.toBinaryString(num1));

        System.out.println(Integer.toOctalString(num1));

        System.out.println(Integer.toHexString(num1));

    }

}

```

```

D:\CDAC\OOPJ\Assignmernt>java data33
111
7
7

```

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`).

```

class data35{

    public static void main(String args[]){

        long num1 = Long.BYTES;

        System.out.println(num1);

    }

}

```

```

D:\CDAC\OOPJ\Assignmernt>java data35
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`).

```
class data36{

    public static void main(String args[]){

        long num1 = Long.MIN_VALUE;

        System.out.println(num1);

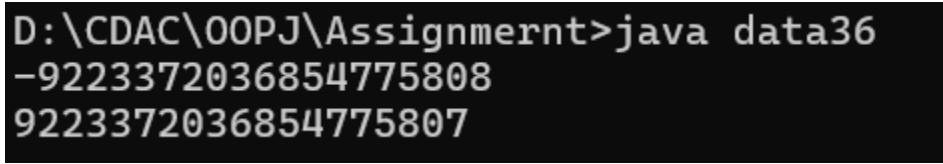
        //System.out.println(num1);

        long num2 = Long.MAX_VALUE;

        System.out.println(num2);

    }

}
```



```
D:\CDAC\OOPJ\Assignmernt>java data36
-9223372036854775808
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)`).

```
class data37{

    public static void main(String args[]){

        long number = 36546546;

        System.out.println(Long.toString(number));

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data37
36546546
```

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)`).

```
class data38{

    public static void main(String args[]){

        String strNumber = "36546546";

        System.out.println(Long.parseLong(strNumber));

    }

}
```

```
D:\CDAC\00PJ\Assignmernt>java data38
36546546
```

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`).

```
class data39{

    public static void main(String args[]){

        String strNumber = "Ab12Cd3";

        System.out.println(Long.parseLong(strNumber));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data39
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Long.parseLong(Long.java:711)
    at java.base/java.lang.Long.parseLong(Long.java:836)
    at data39.main(Assignment3.java:460)
```

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)).

```
class data40{

    public static void main(String args[]){

        long number = 116;

        System.out.println(Long.valueOf(number));

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data40
116
```

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().

```
class data41{

    public static void main(String args[]){

        long number = 116;

        System.out.println(Long.valueOf(number));

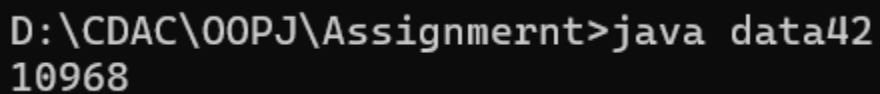
    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data41
116
```

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)).

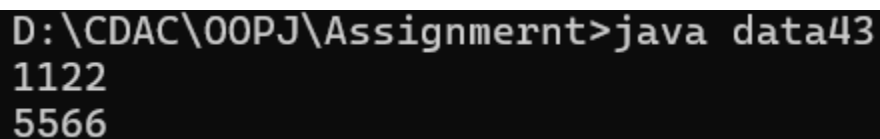
```
class data42{  
  
    public static void main(String args[]){  
  
        long number = 1123;  
  
        long number1 = 9845;  
  
        System.out.println(Long.sum(number, number1));  
  
    }  
  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data42  
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)).

```
class data43{  
  
    public static void main(String args[]){  
  
        long number = 1122;  
  
        long number1 = 5566;  
  
        System.out.println(Long.min(number, number1));  
  
        System.out.println(Long.max(number, number1));  
  
    }  
  
}
```



```
D:\CDAC\OOPJ\Assignmernt>java data43  
1122  
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)`).

```
class data44{

    public static void main(String args[]){

        long number = 7;

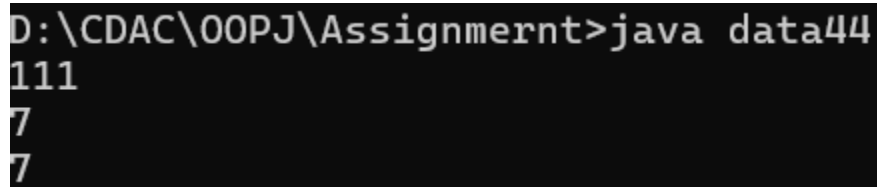
        System.out.println(Long.toBinaryString(number));

        System.out.println(Long.toOctalString(number));

        System.out.println(Long.toHexString(number));

    }

}
```



```
D:\CDAC\OOPJ\Assignmernt>java data44
111
7
7
```

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

```
class data45{

    public static void main(String args[]){

        long number = 70000;

        int num =(int) number;

        double n1=(double) num;

        System.out.println(num);

        System.out.println(n1);

    }

}
```



```
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data45  
70000  
70000.0
```

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`).

```
class data46{  
  
    public static void main(String args[]){  
  
        double n1=Double.BYTES;  
  
        System.out.println(n1);  
  
    }  
  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data46  
8.0
```

- 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`).

```
class data47{  
  
    public static void main(String args[]){
```

```

        double n1=Double.MIN_VALUE;

        double n2=Double.MAX_VALUE;


        System.out.println(n1);

        System.out.println(n2);

    }

}

```



```

D:\CDAC\00PJ\Assignmernt>java data47
4.9E-324
1.7976931348623157E308

```

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)`).

```

class data48{

    public static void main(String args[]){

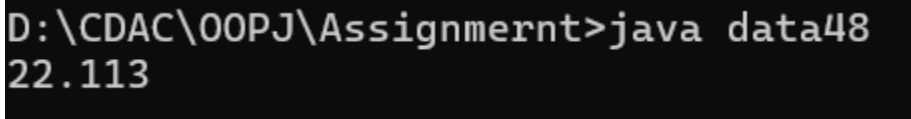
        double number=22.113;

        System.out.println(Double.toString(number));

    }

}

```



```

D:\CDAC\00PJ\Assignmernt>java data48
22.113

```

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)`).

```
class data49{

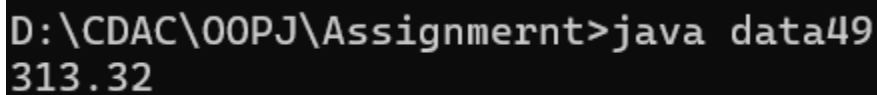
    public static void main(String args[]){

        String strNumber="313.32";

        System.out.println(Double.parseDouble(strNumber));

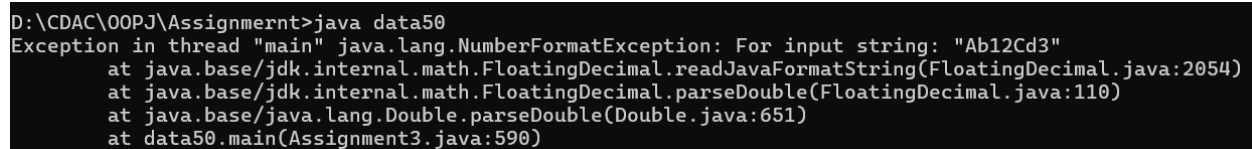
    }

}
```



```
D:\CDAC\OOPJ\Assignmernt>java data49
313.32
```

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`).



```
D:\CDAC\OOPJ\Assignmernt>java data50
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
    at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)
    at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
    at java.base/java.lang.Double.parseDouble(Double.java:651)
    at data50.main(Assignment3.java:590)
```

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)`).

```
class data51{

    public static void main(String args[]){

        double strNumber=111.1313;

        System.out.println(Double.valueOf(strNumber));

    }

}
```

```
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data51  
111.1313
```

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)`).

```
class data52{  
  
    public static void main(String args[]){  
  
        String strNumber = "123.45";  
  
        Double doubleValue = Double.valueOf(strNumber);  
  
        System.out.println(doubleValue);  
  
    }  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data52  
123.45
```

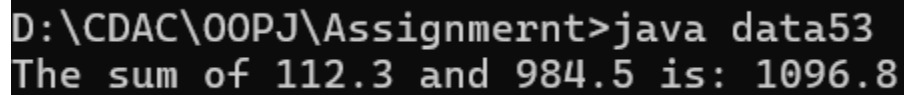
i. Declare two double variables with values `112.3` and `984.5`, and add them using a method from the `Double` class. (Hint: Use `Double.sum(double, double)`).

```
class data53{  
  
    public static void main(String args[]){  
  
        double num1 = 112.3;  
  
        double num2 = 984.5;  
  
        double sum = Double.sum(num1, num2);
```

```

        System.out.println("The sum of " + num1 + " and " + num2 + " is: " + sum);
    }
}

```



```

D:\CDAC\00PJ\Assignmernt>java data53
The sum of 112.3 and 984.5 is: 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)`).

```

class data54{

    public static void main(String args[]){

        double num1 = 112.2;

        double num2 = 556.6;

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

        double maxValue = Double.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);

    }

}

```

```
D:\CDAC\OOPJ\Assignmernt>java data54
The minimum value between 112.2 and 556.6 is: 112.2
The maximum value between 112.2 and 556.6 is: 556.6
```

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

```
class data55{

    public static void main(String args[]){

        double negativeValue = -25.0;

        double squareRoot = Math.sqrt(negativeValue);

        System.out.println("The square root of " + negativeValue + " is: " + squareRoot);

    }

}
```

```
D:\CDAC\OOPJ\Assignmernt>java data55
The square root of -25.0 is: NaN
```

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

```
class data56{

    public static void main(String args[]){

        double num1 = 0.0;

        double num2 = 0.0;

        double result = num1 / num2;

        System.out.println("The result of dividing " + num1 + " by " + num2 + " is: " + result);

    }

}
```

```
}  
  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data56  
The result of dividing 0.0 by 0.0 is: NaN
```

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

```
class data57{  
  
    public static void main(String args[]){  
  
        double originalDouble = 123.456;  
  
        int intValue = (int) originalDouble;  
  
        float floatValue = (float) originalDouble;  
  
        long longValue = (long) originalDouble;  
  
        System.out.println("Original double value: " + originalDouble);  
  
        System.out.println("Double to int: " + intValue);  
  
        System.out.println("Double to float: " + floatValue);  
  
        System.out.println("Double to long: " + longValue);  
  
    }  
  
}
```

```
D:\CDAC\OOPJ\Assignmernt>java data57  
Original double value: 123.456  
Double to int: 123  
Double to float: 123.456  
Double 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()`).
- Then, use the `valueOf` method of the `String` class. (e.g., `String.valueOf()`).

```
class data58{  
  
    public static void main(String args[]){  
  
        int intValue = 42;  
  
        double doubleValue = 3.14159;  
  
        boolean booleanValue = true;  
  
        char charValue = 'A';  
  
        float floatValue = 1.23f;  
  
        long longValue = 123456789L;  
  
        short shortValue = 100;  
  
        byte byteValue = 10;  
  
  
        String intToString = Integer.toString(intValue);  
  
        String doubleToString = Double.toString(doubleValue);  
  
        String booleanToString = Boolean.toString(booleanValue);  
  
        String charToString = Character.toString(charValue);  
  
        String floatToString = Float.toString(floatValue);  
  
        String longToString = Long.toString(longValue);  
  
        String shortToString = Short.toString(shortValue);  
  
    }  
}
```



```
String byteToString = Byte.toString(byteValue);
```

```
String intValueOf = String.valueOf(intValue);
```

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

```
String booleanValueOf = String.valueOf(booleanValue);
```

```
String charValueOf = String.valueOf(charValue);
```

```
String floatValueOf = String.valueOf(floatValue);
```

```
String longValueOf = String.valueOf(longValue);
```

```
String shortValueOf = String.valueOf(shortValue);
```

```
String byteValueOf = String.valueOf(byteValue);
```

```
System.out.println("Using wrapper class's toString method:");
```

```
System.out.println("int to String: " + intValueOf);
```

```
System.out.println("double to String: " + doubleValueOf);
```

```
System.out.println("boolean to String: " + booleanValueOf);
```

```
System.out.println("char to String: " + charValueOf);
```

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

```
System.out.println("long to String: " + longValueOf);
```

```
System.out.println("short to String: " + shortValueOf);
```

```
System.out.println("byte to String: " + byteValueOf);
```

```
System.out.println("\nUsing String's valueOf method:");
```

```
System.out.println("int to String: " + intValueOf);
```

```
System.out.println("double to String: " + doubleValueOf);
```

```
System.out.println("boolean to String: " + booleanValueOf);
```

```
System.out.println("char to String: " + charValueOf);
```

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

```
System.out.println("long to String: " + longValueOf);
```

```
System.out.println("short to String: " + shortValueOf);
```

```
System.out.println("byte to String: " + byteValueOf);
```

```
}
```

```
}
```

```

D:\CDAC\00PJ\Assignmernt>java data58
Using wrapper class's toString method:
int to String: 42
double to String: 3.14159
boolean to String: true
char to String: A
float to String: 1.23
long to String: 123456789
short to String: 100
byte to String: 10

Using String's valueOf method:
int to String: 42
double to String: 3.14159
boolean to String: true
char to String: A
float to String: 1.23
long to String: 123456789
short to String: 100
byte to String: 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).

```

class data59{

    // Instance variables

    byte instanceByte;

    short instanceShort;

    int instanceInt;

    long instanceLong;

    float instanceFloat;

```

```
double instanceDouble;
```

```
char instanceChar;
```

```
boolean instanceBoolean;
```

```
// Static variables
```

```
static byte staticByte;
```

```
static short staticShort;
```

```
static int staticInt;
```

```
static long staticLong;
```

```
static float staticFloat;
```

```
static double staticDouble;
```

```
static char staticChar;
```

```
static boolean staticBoolean;
```

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

```
    data59 example = new data59();
```

```
    System.out.println("Instance variables default values:");
```

```
    System.out.println("byte: " + example.instanceByte);
```

```
    System.out.println("short: " + example.instanceShort);
```

```
    System.out.println("int: " + example.instanceInt);
```

```
    System.out.println("long: " + example.instanceLong);
```

```
System.out.println("float: " + example.instanceFloat);  
  
System.out.println("double: " + example.instanceDouble);  
  
System.out.println("char: '" + example.instanceChar + "'");  
  
System.out.println("boolean: " + example.instanceBoolean);
```

```
System.out.println("Static variables default values:");
```

```
System.out.println("byte: " + staticByte);
```

```
System.out.println("short: " + staticShort);
```

```
System.out.println("int: " + staticInt);
```

```
System.out.println("long: " + staticLong);
```

```
System.out.println("float: " + staticFloat);
```

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

```
System.out.println("char: '" + staticChar + "'");
```

```
System.out.println("boolean: " + staticBoolean);
```

```
}
```

```
}
```

```
D:\CDAC\00PJ\Assignmernt>java data59
Instance variables default values:
byte: 0
short: 0
int: 0
long: 0
float: 0.0
double: 0.0
char: ''
boolean: false
Static variables default values:
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).

```
class Operations {

    public static void main(String[] args) {

        char operator;

        Double number1, number2, result;

        // create an object of Scanner class

        Scanner input = new Scanner(System.in);
```

```
// ask users to enter operator

System.out.println("Choose an operator: +, -, *, or /");

operator = input.next().charAt(0);


// ask users to enter numbers

System.out.println("Enter first number");

number1 = input.nextDouble();


System.out.println("Enter second number");

number2 = input.nextDouble();


switch (operator) {

    // performs addition between numbers

    case '+':

        result = number1 + number2;

        System.out.println(number1 + " + " + number2 + " = " + result);

        break;


    // performs subtraction between numbers

    case '-':

        result = number1 - number2;
```

```
        System.out.println(number1 + " - " + number2 + " = " + result);

        break;

// performs multiplication between numbers

case '*':

    result = number1 * number2;

    System.out.println(number1 + " * " + number2 + " = " + result);

    break;

// performs division between numbers

case '/':

    result = number1 / number2;

    System.out.println(number1 + " / " + number2 + " = " + result);

    break;

default:

    System.out.println("Invalid operator!");

    break;

}

input.close();

}
```



```
D:\CDAC\00PJ\Assignmernt>java Operations
Choose an operator: +, -, *, or /
+
Enter first number
5
Enter second number
6
5.0 + 6.0 = 11.0
```

```
D:\CDAC\00PJ\Assignmernt>java Operations
Choose an operator: +, -, *, or /
*
Enter first number
5
Enter second number
2
5.0 * 2.0 = 10.0
```

