

In this lab, you will learn to use

- reference data type: BigInteger
- loops
- console input and output
- method overloading

**Problem statement:** Write a program that takes an integer input  $n$  from the user and calculates its Factorial. **Factorial rules:**

- $0! = 1$
- $n! = n \times (n-1)!$
- Factorial of a negative number is undetermined. So your program should print 1 whenever a negative number is passed to it.

The list of factorials for  $n \leq 15$  is shown below. Notice when  $n \geq 13$ ,  $n!$  values differ because int can handle only up to  $2,147,483,647^*$ . So you need to create a program that uses BigInteger data type to calculate factorials for  $n \geq 13$ . As shown in Fig.1 to Fig.5, the program should ask for a number input, and then calculate its factorial correctly.

Factorial using int	Factorial using BigInteger
1! = 1	1! = 1
2! = 2	2! = 2
3! = 6	3! = 6
4! = 24	4! = 24
5! = 120	5! = 120
6! = 720	6! = 720
7! = 5,040	7! = 5,040
8! = 40,320	8! = 40,320
9! = 362,880	9! = 362,880
10! = 3,628,800	10! = 3,628,800
11! = 39,916,800	11! = 39,916,800
12! = 479,001,600	12! = 479,001,600
13! = 1,932,053,504	13! = 6,227,020,800
14! = 1,278,945,280	14! = 87,178,291,200
15! = 2,004,310,016	15! = 1,307,674,368,000

**Solution Design:** The program has one class Factorial that has three methods as shown in Figure 6. The calcFactorial() method is overloaded as one takes int and the other BigInteger as a parameter.

#### Instructions:

1. Download Factorial.java and TestFactorial.java from Canvas
2. Create a package named lab2 and import these two files into this package
3. Fill in your code in Factorial.java. Do not change any method signatures.
4. Test your Factorial.java using TestFactorial.java. Do not change TestFactorial.java.
5. Make sure your console output also comes out as expected.
6. Write your Andrew-id and name as top line comment in your **Factorial.java** file only and submit it on Canvas.

```
Enter number
0
Factorial of 0 is 1
```

Figure 1:  $n = 0$

```
Enter number
1
Factorial of 1 is 1
```

Figure 2:  $n = 1$

```
Enter number
5
Factorial of 5 is 120
```

Figure 3:  $n = 5$

```
Enter number
-5
Factorial of -5 is 1
```

Figure 4:  $n = -5$

```
Enter number
13
Factorial of 13 is 6,227,020,800
```

Figure 5:  $n = 13$

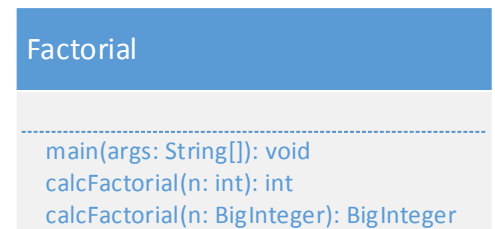


Figure 6: Class diagram

\* Can you find out at what value of  $n$  will  $n!$  run out of space if  $n!$  is stored in a variable of type 'long'?