# Basis of Computer Programming (java)
# Lab Exercise 5

**[Experimental Objective]**

1. Learn how to use the *if* and *if…else* selection statements to choose among alternative actions.
2. Learn how to use the *while* repetition statement to execute statements in a program repeatedly.
3. Learn how to use counter-controlled repetition and sentinel controlled repetition.
4. Learn how to use the compound assignment, increment and decrement operators.

**[Exercises]**

1. Write an application which can convert the grades on 100 point scale into GPA according to the following table.

| Grade | GPA |
|---|---|
| 100~90 | 4.0 |
| 89~80 | 3.0 |
| 79~70 | 2.0 |
| 69~60 | 1.0 |
| 59~0 | 0 |

2. In mathematics, the Fibonacci numbers are the numbers in the following integer sequence, called the Fibonacci sequence, and characterized by the fact that every number after the first two numbers is the sum of the two preceding ones

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \cdots\cdots$$

Write an application that inputs one integer $n$ from the user and output the first $n$ Fibonacci numbers in the Fibonacci sequence.

3. (Book P116, 4.24, Dangling-else Problem) Determine the output for each of the given sets of code when *x* is 9 and *y* is 11 and when *x* is 11 and *y* is 9. The compiler ignores the indentation in a Java program. Also, the Java compiler always associates an else with the immediately preceding if unless told to do otherwise by the placement of braces ({}). On first glance, you may not be sure which if a particular else matches—this situation is referred to as the "dangling-else problem." We've eliminated the indentation from the following code to make the problem more challenging. [Hint: Apply the indentation conventions you've learned.]

```
a)  if ( x < 10 )
    if ( y > 10 )
    System.out.println( "*****" );
    else
    System.out.println( "#####" );
    System.out.println( "$$$$$" );
b)  if ( x < 10 )
    {
    if ( y > 10 )
    System.out.println( "*****"  );
    }
    else
    {
    System.out.println( "#####" );
    System.out.println( "$$$$$" );
    }
```

4. Create a class called GuessingNumber that includes
   - An Integer type attributes --- magicNum;
   - A constructor to initialize the attribute. The attribute should be initialized to a random integer between 0 and 10. The following class Random can be used to generate the random integer.

     ```
     Random random = new Random();
     magicNum = random.nextInt(10);
     ```

   - One method named *guess* which keeps asking the user to input an integer between 0 and 10 until the input number is equal to the attribute magicNum. When the input number is not equal to the attribute magicNum, the method should output "Please try again, input another integer between 0 and 10" and wait for the user to input a new integer . When the input number is equal to the attribute magicNum, the method should output "Congratulations" and terminate.

   Write a test application named TestGuessingNumber to test the program.

5. (Book P115, 4.20, Find the Largest Number) The process of finding the largest value is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each salesperson. The salesperson who sells the most units wins the contest. Write a Java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:
   a) counter: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
   b) number: The integer most recently input by the user.
   c) largest: The largest number found so far.

6. (Book P117, 4.28, Printing the Decimal Equivalent of a Binary Number) Write an application that inputs an integer containing only 0s and 1s (i.e., a binary integer) and prints its decimal equivalent. [Hint: Use the remainder and division operators to pick off the binary number's digits one at a time, from right to left. In the decimal number system, the rightmost digit has a positional value of 1 and the next digit to the left a positional value of 10, then 100, then 1000, and so on. The decimal number 234 can be interpreted as 4 * 1 + 3 * 10 + 2 * 100. In the binary number system, the rightmost digit

has a positional value of 1, the next digit to the left a positional value of 2, then 4, then 8, and so on. The decimal equivalent of binary 1101 is 1 * 1 + 0 * 2 + 1 * 4 + 1 * 8, or 1 + 0 + 4 + 8 or, 13.]

7. (Book P117, 4.34, Factorial) The factorial of a nonnegative integer n is written as n! (pronounced "n factorial") and is defined as follows:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 1 \text{ (for } n \geq 1).$$

and

$$n! = 1 \text{ (for } n = 0).$$

For example, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, which is 120.

a) Write an application that reads a nonnegative integer and computes and prints its factorial.

b) Write an application that estimates the value of the mathematical constant *e* by using the following formula. Allow the user to enter the number of terms to calculate.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots$$

**[Assignments]**

Question 4、5、6、7 of [Exercises]