Introduction to Computer Programming (Java A)
Tutorial 4

[Objectives]

1. Learn how to use the *while, for* repetition statement to execute statements in a program.
2. Learn how to use the *switch* selection statements to choose among alternative actions.
3. Learn how to use the *break, continue, return* statements in a program.

[Exercises]

1. Rewrite the exercise5 in lab3. Use *for* repetition statements to estimate the value of π, according to the specified number of iterations and precision threshold.

Think about this: when to use *for* and when to *while*?

Calculate the value of π from the infinite series

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \cdots$$

(1) Input an integer ***n***, which presents the times of iterations. The estimated value is more precise when ***n*** is bigger.

(2) Input a double value, which presents a precision threshold. The program will terminate when the difference between two successive values is smaller than the precision threshold. Print the value of π, and the number of iterations.

2. Rewrite the exercise 2 in lab3. Use *switch* to calculate the GPA, and then realize following requirements.

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

Write a program to calculate the GPA of a student according to the method used by SUSTech. The program should take as input a student's credit hour and score of each course. The process should continue until the user types -1 as a score. After receiving all inputs, the program should output the final GPA of the student.

Think about this: when *if…else…* could be replaced by *switch*?

Sample output

```
3 95
2 89
3 77
3 67
1 95
-1
final gpa is 2.6
```

3. There are 30 or 31 days in a month except February. There are 28 days in February in a common year, and 29 days in a leap year. Write a program to input year and month by command line and show the days of this month using *switch*.

A year is a leap year if:
(1) divisible by 4, but not divisible by 100;
(2) or divisible by 400;

Please complete the program in *DaysofYearMonth.java*

Sample output:

```
D:\CS102A>java DaysOfYearMonth 2019 3
March of 2019 has 31 days.

D:\CS102A>java DaysOfYearMonth 2019 2
February of 2019 has 28 days.

D:\CS102A>java DaysOfYearMonth 1900 2
February of 1900 has 28 days.

D:\CS102A>java DaysOfYearMonth 2000 2
February of 2000 has 29 days.
```

4. Complete *MultiplicationTable.java* to print the multiplication table.
(1) add *break* and *continue* statements where appropriate; (do not change the conditions of the *for* statements)
(2) Your program should display a multiplication table of a given size in [1, 9]
(3) Your program should terminate if the given size is 0
(4) Your program should warn users when they provide invalid inputs and continue to wait for valid inputs

Sample output:

```
Please input a number to print the Multiplication Table [0 to terminate]:
-4
Please input a number between [1,9]
Please input a number to print the Multiplication Table [0 to terminate]:
1
1 * 1 =  1
Please input a number to print the Multiplication Table [0 to terminate]:
3
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
Please input a number to print the Multiplication Table [0 to terminate]:
9
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
1 * 4 =  4  2 * 4 =  8  3 * 4 = 12  4 * 4 = 16
1 * 5 =  5  2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
1 * 6 =  6  2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
1 * 7 =  7  2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7 = 49
1 * 8 =  8  2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8 = 56  8 * 8 = 64
1 * 9 =  9  2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7 * 9 = 63  8 * 9 = 72  9 * 9 = 81
Please input a number to print the Multiplication Table [0 to terminate]:
0
```

Think about this: can you also use return statements for this task?

Further reading:
How to avoid using break and continue statement?
https://blog.csdn.net/fjian123/article/details/80408539