Lab11
Java Looping Statements and Methods

No Submission is Required

Note

In this lab, you will be given TWO problems to solve using Java. To make the program implementation easier, you are suggested to write your program using **eclipse** instead of doing it directly using paper & pencil. No submission is required for this lab. However, you are highly recommended to spend time on the exercises so that you get a thorough understanding of Java programming using looping statements and methods.

Objective

The objective of this lab is to reinforce your programming concepts, especially Java iterative (looping) statements and methods taught in the last few lectures.

Software Required

The following software is required for this lab.

1. Java Development Kit Version 1.8
2. An Integrated Development Environment, e.g. eclipse

Introduction

In this lab, you will practice how to solve problems using iterative statements. For instances, while loop, for loop and do-while loop. In addition, you will be ask to implement programs using methods. As usual, you will be given a brief review on these topics before going through the details of the lab exercise.

(Note: You are suggested to spend more time on the lab exercise. Since that should give you a thorough revision on what you should have learnt in class.)

Background

1. Executing statements repeatedly using looping statements:

Syntax:

```
// If <boolean expression> is evaluated to true, <statement> is executed and this
// process is repeated until the <boolean expression> is evaluated to false.
while(<boolean expression>)
   <statement>;

// If <boolean expression> is evaluated to true, <statement 1>, <statement 2>,
// …,    are executed and this process is repeated until the <boolean expression> is
// evaluated to false.
while(<boolean expression>)
{
   <statement 1>;
   <statement 2>;
   // …
}

// <initialize> is a statement to initialize the control variable, <boolean expression>
// is an expression returns a boolean result, <update> is a statement to update
// the control variable, <statement> denote program statement need to be executed when
the <boolean expression> is evaluated true.
for (<initialize>; <boolean expression>; <update>)
   <statement>;

// <initialize> is a statement to initialize the control variable, <boolean expression>
// is an expression returns a boolean result, <update> is a statement to update
// the control variable, <statement 1>, <statement 2>, … denote program statement need to
be executed when the <boolean expression> is evaluated true.
for (<initialize>; <boolean expression>; <update>)
{
   <statement 1>;
   <statement 2>;
   …
}

// <boolean expression> is an expression returns a boolean result, <statement> denote
// statement need to be executed when <boolean expression> is evaluated to true.
do
   <statement>;
while(<boolean expression>);

// <boolean expression> is an expression returns a boolean result, <statement 1>,
// <statement 2>, … denote statement need to be executed when <boolean expression> is
// evaluated to true.
do {
   <statement 1>;
   <statement 2>;
…
}while(<boolean expression>);
```

2. Modular programming using methods

Syntax:

```
// <return type> is the type of data that will be returned by the method, <method name> is the
// name of the method, <parameter list> is a list of variables used to receive inputs from calling
// method, <statement 1>, <statement 2>, … are statements to be executed in the mehod
public static <return type> <method name>( <parameter list>)
{
   <statement 1>;
   <statement 2>;
   <statement 3>;
   // …
   <statement n>;
}

// Method can be called (or invoked) with its name together with a list of values / variables in the
// argument list
<method name>( <argument list>)
```

Example:

MethodExample.java

```
public class MethodExample
{
  // A method to compute a to the power b
  public static int power(int a, int b)
  {
    int result = 1;
    for(int i=0; i<b; i++)
       result *= a;
    return result;
  }

  public static void main(String[] args)
  {
    int number = 5;
    int exponent = 4;
    int ans = power(number, exponent);
    System.out.println("Answer is " + ans);
  }
}
```

Lab Exercise

In this lab, you will be asked to solve **TWO** problems using Java. Please refer to the problem specification below for the requirements.

Questions:

1. Write a Java program "MultiTest.java" with the use of method to test students' ability to do multiplication. The method should ask student 5 multiplication questions one by one and checks the answers. The method prints out the number of correct answers given by the student. The method random() from the Math class of the Java library can be used to produce two positive one-digit integers (i.e. 1,2,3,4, …) in each question.

   The expected output of your program should be as follows:

   Sample output:

   ```
   How much is 6 times 7? 42
   How much is 2 times 9? 18
   How much is 9 times 4? 36
   How much is 4 times 2? 8
   How much is 3 times 1? 2
   4 answers out of 5 are correct.
   --------------------------------
   ```

   The header of the method that you have to implement is as follows:
   ```
   public static void mulTest()
   ```

2. Write a Java program "DigitsExtraction.java" with the use of method to extract the odd digits from a positive number n and combines the odd digits sequentially into a new number. The method should accept a long integer and return a new number back to the calling method. If the input number does not contain any odd digits, then returns -1. Note that your program should allow user to repeatedly input positive numbers for odd digits extraction until the input number is -1.

   The expected output of your program should be as follows:

   Sample output:

   ```
   Input an integer: 12345
   Extracted odd digits: 135
   Input an integer: 54123
   Extracted odd digits: 513
   Input an integer: 246
   Extracted odd digits: -1
   Input an integer: -12
   Invalid input
   Input an integer: -1
   ```

   The header of the method that you have to implement is as follows:
   ```
   public static long extractOddDigits(long n)
   ```