

## Homework Assignment #2

**Assigned: Thursday, July 05<sup>th</sup>, 2012**

**Due: 11pm on Monday, July 09<sup>th</sup>**

**Instructor: Yinggang Huang (email: [huang58@wisc.edu](mailto:huang58@wisc.edu))**

**TA: Newsha Ardalani (email: [newsha@cs.wisc.edu](mailto:newsha@cs.wisc.edu))**

### Hand in your homework:

This homework assignment covers concepts learned in Chapters 3(decision) && 4(loops). You must do this homework alone. Submit your HW to dropbox created for HW2 in Learn@UW.

Assignment grading questions must be raised with Newsha or me within one week after the assignment is returned.

### Collaboration Policy:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

**Problem 1: General Concepts (8 points)**

**Find the errors in each of the following segments of code:**

- I. What is wrong with the following while repetition structure?

```
while ( x>=0)
    sum += x;
```

- II. The following code should print whether integer value is even or odd:

```
switch (value%2) {
case 0:
    System.out.println( "Even integer");
case 1:
    System.out.println( "Odd integer");
}
```

- III. The following code should print the values 1 to 10:

```
n = 1;
while (n < 10)
    System.out.println(n++);
```

- IV. The following code should output the odd integers from 19 to 1:

```
for ( i = 19; i >=1 ; i+=2 )
    System.out.println( i);
```

**Problem 2: Program I (7 points)**

**Pythagorean Triples:** A right triangle can have sides whose lengths are all integers. The set of three integer values for the lengths of the sides of a right triangle is called a Pythagorean triple. The lengths of the three sides must satisfy the relationship that the sum of the squares of two sides is equal to the square of hypotenuse.

Write an application to find all Pythagorean triples for side1, side2 and the hypotenuse, all no longer than 500. Use a triple nested for loop that tries all possibilities. This method is an example of “brute force” computing. You will learn in more advanced computer science courses that there are large number of interesting problems for which there is no known algorithmic approach other than using sheer brute force.

**Problem 3: Program II (15 points)**

## Calculator mini-project

People use digital calculator very often if not every day. Modern calculators can do lots of things for standard, scientific, programming, statistics, unit conversion and many many more. Those calculators are actually pieces of software programs written by some programming languages.

Wouldn't it be awesome if we can not only enjoy the convenience that those calculator apps bring to us, but create one on our own? Don't get intimidated, trust ourselves, we can manipulate programming after we learned it. You'll be surprised about how much the customers can shell out for the app you write if it's really cool.

Write a Java application that provides a simple command-line calculator that supports addition, subtraction, multiplication, and division. Multiple operators can be processed in a single expression.

Valid expressions are of the following form: NUM OP NUM OP ... NUM =, where NUM represents a decimal number and OP is either a +, -, \*, or /. Spaces are required to separate the numbers from the operators. The equals sign at the end is required to indicate to the program that it should process the expression and return the result.

For simplicity, this program does NOT support proper order of operations (or parenthesization). Rather, operations are processed from left to right, regardless of normal precedence rules. It turns out that good order of operations is a lot more difficult to implement and requires more knowledge in programming than what we've learned so far.