Concordia University
Department of Computer Science
and Software Engineering

Introduction to Programming
COMP 248 --- Winter 2006 --- Section U

Assignment #3

Deadline:	week 8, Friday March 3rd, 2006
Evaluation:	3% of final mark
Late submission:	not accepted
Teams:	this is an individual assignment
Purpose:	to help you practice the material presented in Chapter 3

Exercises

1. The Babylonian algorithm to compute the square root of a number ***n*** is as follows:

    1. *Make a guess at the answer (you can pick **n/2** as your initial guess)*
    2. *compute **r = n/guess***
    3. *set **guess = (guess+r)/2***
    4. *go back to step 2 for as many iterations as necessary. The more you repeat step 2 and 3, the closer guess will become to the square root of **n**.*

    Write a program that inputs two integers ***n*** and ***i***, and iterates through the Babylonian algorithm ***i*** times, and outputs the answer as a **`double`** to two decimal places, including the guess result at each iteration. The algorithm should also stop if ***`guess`*** reaches the square root within the desired number of iterations. For example, two possible sample dialogs might be:

```
Enter a number to compute the square root for: 123123
Enter the number of iterations desired: 123

Square root is:                  350.89
First guess :                  61561.50
guess #2 :                     30781.75
guess #3 :                     15392.87
guess #4 :                      7700.44
guess #5 :                      3858.21
guess #6 :                      1945.06
guess #7 :                      1004.18
guess #8 :                       563.40
guess #9 :                       390.97
guess #10 :                      352.94
guess #11 :                      350.89
I got it in 11 guesses
```

```
Enter a number to compute the square root for: 123123
Enter the number of iterations desired: 5

Square root is:                  350.89
First guess :                  61561.50
guess #2 :                     30781.75
guess #3 :                     15392.87
guess #4 :                      7700.44
guess #5 :                      3858.21
guess #6 :                      1945.06
I did not get it yet, need more iterations
```

2. *In the game of craps, a pass line bet proceeds as follows. Two six-sided dice are rolled; the first roll of the dice in a craps round is called the "come out roll". A come out roll of 7 or 11 automatically wins, and a come out roll of 2, 3 or 12 automatically loses. If 4,5,6,8,9, or 10 is rolled on the come out roll, that number becomes the point. The player keeps rolling the dice until either 7 or the point is rolled. If the point is rolled first, then the player wins the bet. If a 7 is rolled first, then the player loses.*

   Write a program that simulates a game of craps using these rules without human input. Instead of asking for a wager, the program should calculate whether the player would win or lose. The program should simulate rolling the two dice and calculate the sum. Add a loop so that the program plays 10,000 games. Add counters that count how many times the player wins, and how many times the player loses. At the end of the 10,000 games, compute the percentage winning games and output this value.

   Note: To generate a random number *x*, where *0 <= x <= 1*, use `Math.random();`. For example, multiplying by 6 and converting to an integer results in an integer that is between 0 and 5.

3. Write another version of your game of craps that enables the player to interact with the game in the following manner:

```
Welcome to craps

Come out roll: 5 and 3 : the point is 8
Press enter to roll
Your roll is : 4 and 2 : 6
Press enter to roll
Your roll is : 6 and 2 : 8 you win
Another game? Y

Come out roll: 3 and 4 : 7 the house wins
Another game? Y

Come out roll: 1 and 2 : 3 you win
Another game? Y

Come out roll: 4 and 5 : the point is 9
Press enter to roll
Your roll is 3 and 5 : 8
Press enter to roll
Your roll is : 6 and 1 : 7 the house wins
Another game? N

You won 2 and lost 2
```

Evaluation Criteria (for each individual exercise)

Comments: 6 pts	
Description of the program (authors, date, purpose):	2 pts
Description of variables, constants and algorithm:	4 pts
Programming style: 8 pts	
Use of constants where necessary:	2 pts
Use of significant names for identifiers:	2 pts
Indentation and readability:	2 pts
Simplicity of the algorithm:	2 pts
Results on test data and on unseen data:	6 pts
Total	**20 pts**

Submission Procedure

When you are finished all programs, you should submit a paper version and an electronic version of your java source files. **Both** the paper **and** the electronic submission must be on time; otherwise, the assignment will be considered late.

1. **Paper submission:**
   a) Fill out an expectation of originality form (given to you in class or available on the Web page).
   b) Print all your source code (the `.java` files).
   c) Print a screen shot of your output with the data given above.
   d) Staple everything together and give it to your instructor in class. If you cannot give the assignment in class, go to the Department of Computer Science and Software Engineering secretary desk in EV3.139, ask the receptionist to write the date and time on your assignment and initial it, and put the assignment in your instructor's assignment submission box, located in EV3.414.
2. **Electronic submission:**
   Create one zip file, containing all source files for your assignment. The zip file should be called `assignment#_studentID`, where `#` is the number of the assignment and `studentID` is your student ID number. For example, for the first assignment, student 123456 would submit: assignment1_123456.zip. Use the Electronic Submission Form to upload your zip file. Go to the following URL, fill-in the form, and submit your file: `https://eas.encs.concordia.ca/eas/authentication.jsp`,