CSCE 121 [506–509] Introduction to Program Design and Concepts
Spring 2012 – Hyunyoung Lee
Team Project 1

Assigned on Monday February 20, 2012

Part I: Electronic submission due at 23:59, Tuesday, February 28, 2012

Part II: Electronic submission due at 23:59, Tuesday, March 6, 2012

(hardcopies are always due at the following lab)

(Print this page as a cover sheet for your printouts)

"On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment."

Section: CSCE 121–____________

Team member names:	Signature of student:
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________

NOTE 1 (Teaming up): This is a team project. You will work in a team of two persons. If the number of students in your section is an odd number, then there will be one team with three persons. Within the first two days (Monday–Tuesday, February 20–21), you should find your team-mate (you can use the lab time to discuss this) and inform the TA your team members. If you have not teamed up by 5p.m. Tuesday, the TA will randomly assign you a team.

NOTE 2 (Importance of "team project"): Remember this is a team project. Your team-code should compile and run correctly *as a whole* to receive credits.

NOTE $2\frac{3}{4}$: The next team project will use the code developed in this team project, thus make sure you complete this project and understand it fully.

NOTE 3: Please follow your TA's directions for submitting your assignment through CSnet. Only assignments submitted to CSnet will be graded. Make a printout of each source file and staple it behind this cover sheet. Sign it and bring it to your TA in lab. **If you do not turn in a signed cover sheet your work will not be graded.**

NOTE 4: Your programs will be graded using those you submit on CSnet, not the hardcopy, on unix.cse.tamu.edu using g++. Whatever platform you use to develop your programs, it is your responsibility to make sure your programs also compile and run correctly on unix.cs.tamu.edu using g++.

The grade for this assignment will be based on the following criteria:

- correctness (e.g., adhering to the problem specification and requirements, producing the correct output for all test cases and properly handling input errors and exceptions),
- style (e.g., divide-&-conquer design using functions and classes whenever appropriate, programmer-defined identifiers, appropriate comments, and neatness of logic), and
- syntax (no compilation or link errors).

You will earn total 100 points.

Part I.

1. (20 points) Ch.6 Drill problems 1–5 on page 215. Download the file *calculator02buggy.cpp* from `http://www.stroustrup.com/Programming/calculator02buggy.cpp`.

   For problem 5, include the team member names in the greeting.

   Turn in one final team-code on CSnet. Put as part of head comment, who worked on which parts, for example, a possible way of dividing the work of this part between two team members could be that one person finds two errors in problem 1, and works on problems 2 and 4, and the other person finds the third error in problem 1, and works on problems 3 and 5.

2. (20 points) This part is performed *individually*. While working on the Ch.6 Drill problem 1, try the following input expression, working the control flow by hand:

   $$5 + (6 * 2)/(8 - 3) =$$

   Write an *individual* report on the flow of control showing how the expression is evaluated (starting from `main()`, tracing the corresponding function calls) until the final value is output. Type in your report and turn-in a .txt file on CSnet.

Part II.

1. Ch.7 Drill problems on page 250. Download the file in question from `http://www.stroustrup.com/Programming/calculator08buggy.cpp`. Note the following updates to the grammar:

```
calculation:
    statement
    print
    quit
    calculation statement

statement:
    declaration
    expression

declaration:
    "let" name "=" expression
```

```
expression:
    term
    expression "+" term
    expression "-" term

term:
    primary
    term "*" primary
    term "/" primary
    term "%" primary

primary:
    number
    "(" expression ")"
    "sqrt(" expression ")"
    "pow(" expression "," expression ")"

number:
    floating-point-literal
```

- (15 points) Steps 1–5
- (15 points) Step 7–8
- (15 points) Step 9

2. (15 points) Ch.7 Exercise problem 6 on page 252. Change the command to be `help` instead of `H` or `h`. When the user enters `help`, then display what operators are available and how to print and exit (as you did in Part I, Drill problem 5).

3. Turn in one final team-code for Part II on CSnet. Put as part of head comment, who worked on which parts.