Laboratory 2: Programming Basics and Variables

Lecture notes:

1. A quick review of *hello_comment.c*
2. Some useful information
    a. name your program with extension .c
    b. use –o option to specify the name of the executable. For example, gcc hello.c –o hello
    c. "main ()" used in the book is the same as "int main(void)" we used in class in the hello_comment.c
3. Comment:
    a. Single line comment
    b. Multiple line comment
    c. Where comments should be added
        i. At the very beginning of the program
        ii. Variable declaration
        iii. Function declaration and implementation
        iv. Non-trivial (small) block of statements
        v. Code segment added for testing/debugging purpose
4. functions and variables: two components that make a program
5. Variables
    a. Variable declaration: <type variable_name;>
    b. Basic data types in C: char, int, float, double
    c. Variable name:
        i. Any letter, digits, and _ (underscore) can be in a variable name.
        ii. The first character must be letter or _.
        iii. Variable name should have less than 31 characters.
        iv. Variable names are case sensitive.
        v. The following keywords are reserved: auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while
    d. Tips on naming variables:
        i. Give meaningful names, but not too long (you need to type them)
           radius, color, exam_1, proj_2, StudentID.

      ii. Use upper case letters and _ (occasionally, e.g., between words) to increase readability

abadexofvarname, AGoodVarNameEx, a_good_var_name_ex.

      iii. Add a short comment:

float radius; // radius of the circle, read from input

      iv. Assign safe initial values if possible.

float sum = 0; // the sum of my three exam scores

int max = 0; // the maximum among some integers. Note that this

            // this initial value is not safe.

  e. Assign values to variables:

      i. Direct assignment: *radius = 1.5;*

      ii. Declare variable and assign initial value: *float sum = 0;*

      iii. Result of some computation: *area = 3.14 * radius * radius;*

      iv. Read in from input (see below): *scanf()*

6. Define constants:

  a. Why we need to declare constant as a variable (instead of simply using its value)?

  b. Two ways to define a constant variable in C:

const float PI = 3.14159;

const int SIZE = 85;

#define PI 3.14159

#define SIZE 85

  c. Pay attention to the differences between these two constant variable declarations.

7. Basics on printf() and scanf():

  a. Print out a text message: *printf("Hello, world!"):*

  b. Read in a value   *scanf("%f", &radius);*

  c. Print out the value of a variable   *printf("%f", radius);*

  d. *%: format indicator*. It tells functions such as scanf() and printf() what is the data type of the variable to be expected.

%f for variable of type float and double

%d or *%i* for variable of type int (d stands for decimal integer)

%c for variable of type char

8. circle.c

Lab Description

1. Objectives:

   a. Write very simple programs to practice variable declaration, reading input from standard keyboard and performing simple arithmetic operations..

   b. Get familiar with UNIX and its text editors (vi and/or emacs).

2. Pre-lab preparations: Reading assignments are for our textbook (*Applied C*, Fischer *et al.*)

   a. Reading: Chapters 1 and 2 and Sections 3.1 through 3.5 of Chapter 3

   b. Read the codes posted for week ; namely, bugs.c and circle.c

3. In-lab description:

Lab Report

1. Debug the following code, first by observation and then verify it on the computer (by modifying the code, compiling it, and executing it). Enumerate the bugs that you find and correct them. You can correct on the original code. (These code are posted in the blackboard, bugs.c in the documentation section, for your convenience. Remember to comment out unnecessary parts and check file name.)

    a.

    ```
    include <stdio.h>
     int main(void)
    {printf(Wednesday\nSep. 8, 2010\n");
      return 0;
    }
    ```

    Bugs that you find in the program:

    Which one(s) you did not find?

    b.

    ```
    #include <stdio.h>
    int main(void)
    {print(Wednesday\nSep. 8, 2010\n");
      return 0,
    }
    ```

    Bugs you find in the program:

    Which one(s) you did not find?

    c.

    ```
    #include
    printf( "Wednesday /n  " );
    printf( "Sep. 8, 2010/n");
     return 0;
    ```
    Bugs you find in the program:

    Which one(s) you did not find?

d.

#Include <stdio.h>

Int Main()

{ Printf("Wednesday \n");

   Printf("September 8, 2010\n");

   Return 0;

}

Bugs you find in the program:

Which one(s) you did not find?

e.

 # include <stdio.h>

Int main (void)

{ printf("Wednesday /n")

 printf("Sep. 8, 2010 /n");

  return 0;

}

Bugs you find in the program:

Which one(s) you did not find?

Total number of bug(s) that you have missed from codes a-e:_____

2. Where do you expect the following three statements on variable declarations and what are the
   differences among them?
   1) *float PI_1 = 3.1416;*
   2) *const float PI_2 = 3.1416;*
   3) *#define PI_3 3.1416*

Read the file *comment.txt* in the documents section on blackboard. Think about how you would like to comment one line, multiple lines, etc. and use your style consistently in the programs for questions 3-5 (and hopefully for all your programs in the future. Of course, your style might change from time to time as you learn more and more in programming.)

3. Write a complete C-code (including the documentation) which asks the user to input the length and width of a rectangle, and outputs its area and perimeter, assume both length and width are integers.

4. Write a complete C-code (including the documentation) to compute a student's average grade over 3 projects. The individual project scores range from 0.0 to 100.0.

5. Modify the hello_comment.c code so it asks the user which section he/she is in and then prints out the following : (e.g. on input 0101) (note, it will be a little tricky to print out the first 0 in 0101. But there are a couple of ways to do that.)

    Hello, welcome to ENEE 140 section 0101.