SENG265: Software Development Methods (Summer 2019)

Lab 03 – Git

Week of May 27th

Authors: Nirav Galani*

*based on material provided by Prof. Mike Zastre

This week

- Practice writing, executing and debugging .c files

Creating and running a C file

- Download programA.c from connex
  - Compile
    - $ gcc programA.c –o programA
  - Run the executable
    - $ ./programA

pythag01.c

- Create a C program pythag01.c
  - It will calculate the hypotenuse of a right angle triangle, given the 2 sides at a right angle with each other
  - #include both <stdio.h> and <math.h>
  - Write a main function
    - Declare and assign 2 variables, each representing the 2 right angle sides of a right angle triangle.
    - Write the code to calculate the hypotenuse. (Hint: use sqrt or pow)
    - Print the answer. For example
      - “Right angle triangle with right angled sides of length 3 an 4 has a hypotenuse of 5”
  - Program should work for any lengths of the right angled sides
  - Note that everything is done in the main.

pythag02.c

- Create a C program pythag02.c
  - It will also calculate the hypotenuse of a right angle triangle given the 2 sides at right angle with each other
  - Write a main function similar to pythag01.c with the below modification
  - Write a function named “hypotenuse(……..)” which does the calculation of the hypotenuse. Use parameters as required.
  - Print the answer using a print statement in the main. For example
    - “Right angle triangle with right angled sides of length 3 an 4 has a hypotenuse of 5”
  - Program should work for any lengths of the right angled sides

pythag03.c

- Create a C program pythag03.c
- This will modify pythag01.c to include the following
  - Add to the code → #include <stdlib.h>
  - Lengths of the 2 right angle sides will be entered at the command line when running the executable file. An example of the commands is
    - $ gcc pythag03.c –o pythag03
    - $ ./pythag03 3 4
    - This will execute the program, which will calculate the hypotenuse using the 2 numbers given in the command line and then print the answer to console.
  - If incorrect number for arguments are entered at the command line when running the executable file, the program should print an error message and terminate, For example
    - $ ./pythag03 1
    - This should give an error message and the program should then terminate.

Use Git

- Store all of your lab 3 work in your course remote repository.

Program consisting of more than one file

- Download program01.c and program02.c from connex
- $ gcc program01.c program02.c –o combined
- This creates one executable from both program01.c and program02.c
- Notice that program01.c contains the main and program02.c contains the function expo().
- $ ./combined

q_factorial.c

- Download q_factorial.c
- This program will take a number from the command line, calculate and then print out its factorial.
- Complete the section for command line input checking and calculation of factorial.
- Compile and run the program. What is the factorial of the following numbers
  - 4
  - 13
  - 14
  - 17

q_factorial.c

- The result of factorial for 17 is negative because the memory size allocated for int is too small for the correct answer.
- Change the data type of variable factorial in q_factorial.c to long int.
- Compile and run the program. What is the factorial of
  - 17
  - 30
- Experiment with different data types for the variable factorial and see what result you get.

random_numbers.c

- Download, compile and run random.c
- This program generates random numbers between 0 and 10
- Understand how this code works. You will need to generate random numbers in the next program

sq_cir.c

- Create a new program sq_cir.c
- Consider a square of size 2 X 2
- Assume that the coordinates of the point at the intersection of the diagonals of this square is origin coordinates (0,0)
- Assume a circle drawn with center at (0,0) of radius 1

sq_cir.c

- Calculate the value of π using the ratio of the area of the circle and the square by generating random points
- Generate a large number of points (x1, x2) inside the square using random number generation as demonstrated in random_number.c
- If ($x1^2 + x2^2$) <= 1, then the point is on or inside the circle. Keep count of the total number of points on or inside the circle.
- You can get an approximate value of π with this calculation
  - $4 * \frac{number\ of\ points\ on\ or\ inside\ the\ circle}{number\ of\ points\ inside\ the\ square}$

Git

- Place all you work into your course remote repository