First Assessment (Monday, September 10)

# Things to know Coming In

* Variables
* Conditions
* Loops
* Functions

# C Programming Intro

* C Extensive Library Structure
  + <stdio.h>
  + h means header file
  + What is inside stdio.h?
    - Standard input/output header file
* Main function
  + Required to run code
* C is a compiled language
  + Generates executable
* Programs
  + First Program
    - Find GCF of two positive integers
      * Brute force
      * **Euclid’s Algorithm**
* Functions
  + Printf(string of input);
    - %d or %i
      * Signed decimal integer(-32,767 – 32,767 or -2^31-1 to 2^31-1)
    - %c
      * char
        + -127 to 127
        + 0 to 256
    - %hi
      * Short
    - %u
      * Unsigned int
    - Digital Notation(%f or %F) Scientific Notation(%g, %G, %e, %E, %a, %A)
      * Float
    - %If, %IF, etc.
      * Double
  + Scanf(“%d”,&integer)
    - Scanf(“%s”,&string)
    - Scanf uses pointers because C passes variables by value and we need the function to alter the variable
* Function Prototype
  + The declaration of a function before the code is put in later
  + Names of variables do not matter only the datatype matter
  + Often standardized
    - Python/UOFT docstring
    - Java javadocs
* **NO CODING ON FIRST TEST**
* Euclid’s Algorithm
  + Determine the GCD
  + Mathematical
    - 270 = 192 \* 1 + 78
    - 192 = 78\*2 + 36
    - 78 = 36 \* 2 + 6
    - 36 = 6 \* 6 + 0
    - GCD = 6
  + Programming
    - 270 – (192\*1) = 78
    - 192 – (78\*2) = 36
    - 78 – (36\*2) = 6
    - 36 – (6\*6) = 0
    - 0 – (6\*0) = 0
    - GCD = 6
  + Determining the multiple
    - Brute force multiply a number by increasing intervals until the result is equal to or greater than the other number.
    - Simply divide the numerator by the denominator
  + Calculating the GCD can be improved
    - Replace the multiple function with the modulus operator. The remainder is equal to numerator % denominator

**Study for 2nd Test September 17th**

* No code on next test