Assignment 3

Name: Edwin Rubio

Understanding:

The concept of understanding the problem is a major part in order to accomplish a solution to resolve it. The problem is often is often referred as specification requirements. The process is then broken down into smaller pieces, also known as stepwise refinement, after understanding the requirements. The problem that I encountered was to create a program game that ask the first user to enter an integer for the other user to guess. The second user will have unlimited attempts in order to guess the number, and the program will assists the second user by prompting if to guess higher or lower for the next guess. After the second user has guessed the number, then it prompts the user how many tries it took. This problem was broken down into three sub-problems:

1. Prompt the first user to enter the number to be guessed, and store the number in an int variable.
2. Prompt the second user to guess the number, and store this number in a separated int variable.
3. Take the guess, from the second user, int variables to go through some “if” condition statements and prompt the following:
    a. Guess a “higher” number if “number to be guessed number” is greater than the second user “guessing number”.
    b. Guess a “lower” number if “number to be guessed number” is less than the second user “guessing number”.
    c. If the “number to be guessed” equals to the “guessing number” it prompts the user “you guess it in x tries”.

All of these requirements were then translated into C++ language which included using cout, cin, do/while loops, if conditional statements, and an accumulator variable.

Testing Plan:

One of the tests did not go as expected. The test consisted to determine if the accumulator that stores the number of tries was counting correctly. After performing the test I noticed that the accumulator did not count the last try that had the correct guess. For example, it displayed three tries which actually it was four tries. To fix this issue was to move the accumulator statement at the beginning of the do/while loop. To avoid this in the next test plan will be to create it at the same time while learning the problem (requirements) in order to have a more complete test.

Design:

My initial design was to use a while loop only, and noticed that after the first user entered the number to be guessed it did not prompt the second to try to guess the number. Then I looked into the do/while loop and saw that it was a perfect match on what I needed in order to prompt the second user to guess the number. The do/while loop run the contents inside the loop first which prompts the second user to enter a number.

Implementation:

There were two problems that I encountered using the C++ syntax. The first was when implementing cin. I was adding “>> endl;” at the end of the statement which caused multiple errors when compiling. I referred back to the lectures and found out that it was not needed, since the user will press the “enter” button which is the same as endl. The other problem was the syntax on the do/while loop, since I was not sure where the curly braces and semi-colons were to be entered, so I referred to the following web link, http://mathbits.com/MathBits/CompSci/looping/dowhile.htm which explained the syntax.

Improvement:

During the next project assignments I will improve my problem solving skill by working on the test plan and reviewing the requirements at the same time. That way I don’t forget any important details to implement and to test.