William George

Report for Assignment 3

CS 161 Fall 2014

Week 3 October 15, 2014

Understanding:

This week is all about practicing strings, outputting random numbers, understanding the “for” loop and understanding input validation. The purpose of all the exercises are to help us understand what will be needed for the final project for this week, the random word generator. I’d like to break down each project individually.

strings.cpp

The purpose of strings was to learn the way to read a string word by word and to learn how to use a “for” loop with a string to check the contents. In this exercise we also learned how to ignore an input and how to evaluate both a word and a phrase to validate its contents.

name.cpp

The purpose of this program is to get a better understanding of how to evaluate a string input to parse the string into its individual words. For this project you can use cin or getline to get the input, then you must evaluate it so the output can either have a middle initial or not depending on whether the user inputs a middle name or not.

randNum.cpp

The purpose of this program is to understand the rand function and how to set said function so the pseudorandom numbers can appear to be more random. Also this exercise gives us more practice with the “for” loop.

loop.cpp

The purpose of this program is to compare the individual characters of two strings using a loop. There is an easier way to check and compare the components but at this point we are doing it the hard way with the loops. To compare the two words letter by letter we will likely need to use nested “for” loops.

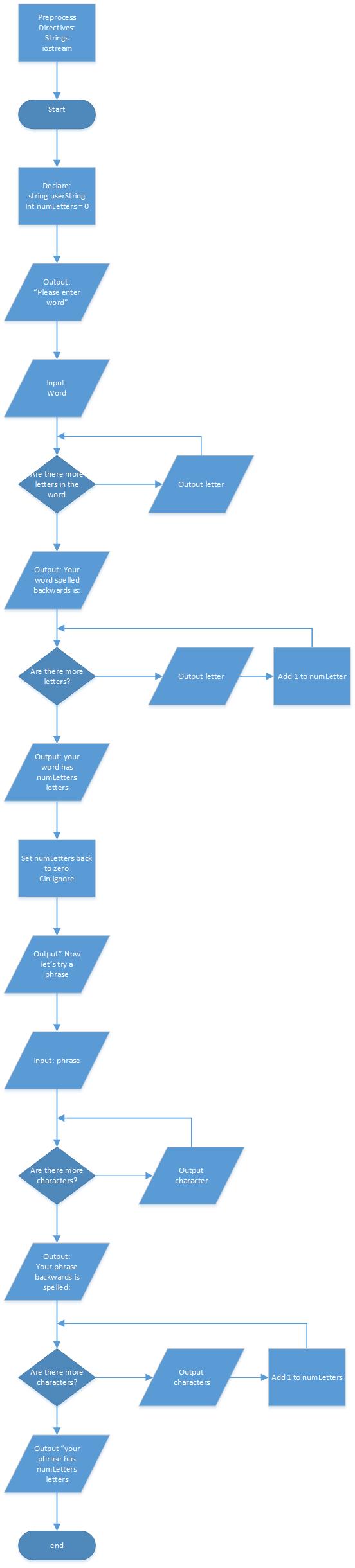
randNum2.cpp

The second random number program is a bit more involved than the first. The program allows the user to input two values from which it will generate a range in which the random number will be chosen from. There is an input validation requirement part of this problem to start to understand how to validate user inputs so the program does what we want even if something wrong is entered by the user.

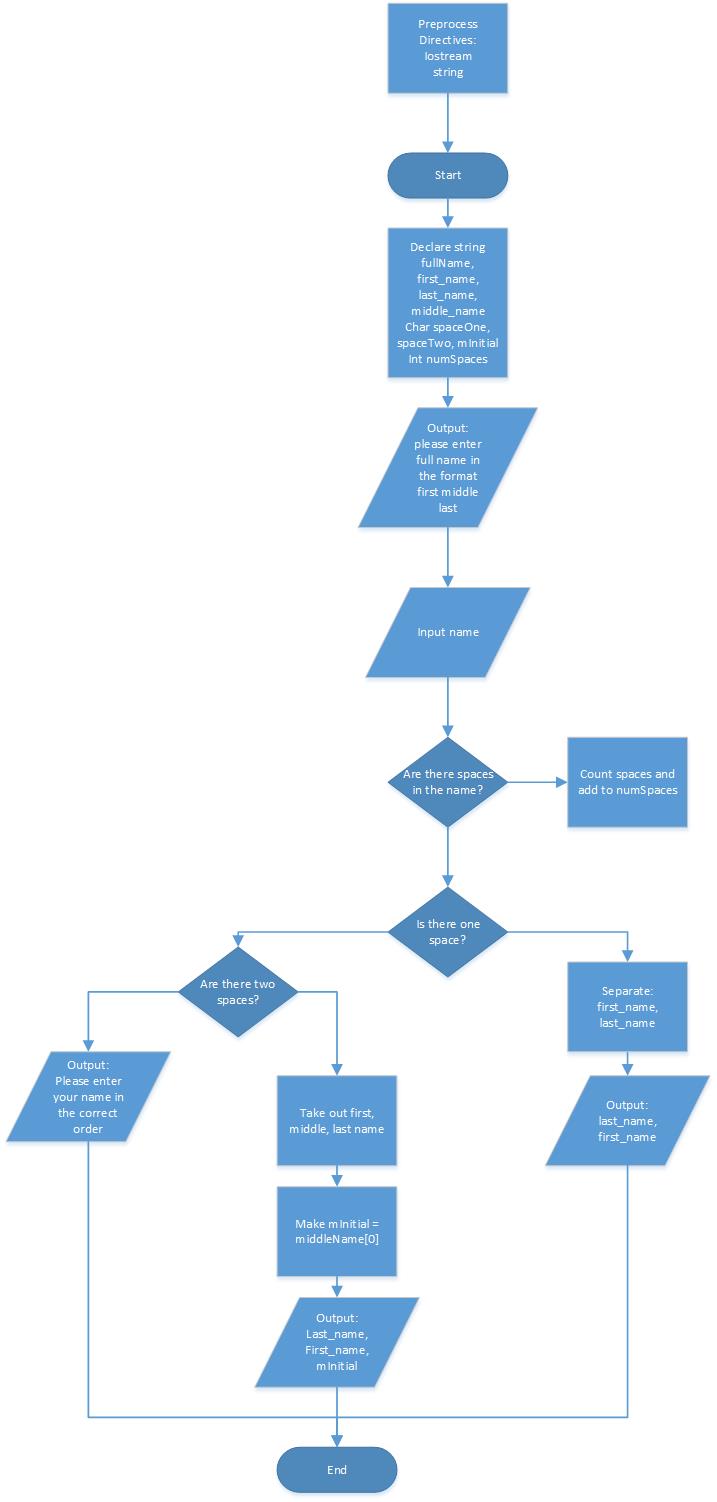
Week 3 Project – Word Guessing Game

The week three project has two parts, the first part being a discussion or drawing of the game and the second being the actual planning of the project and its deliverables. The lessons we’ve learned in the previous assignments should be valuable in the creation of this program. I believe the purpose of this exercise this week is to show to plan for a larger project properly so that it fulfills the requirements set forth by a teacher and later on an employer.

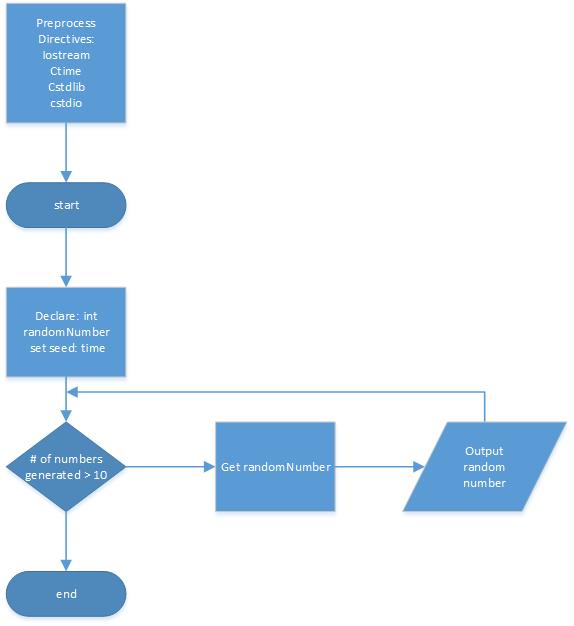
Design

strings.cpp flowchart

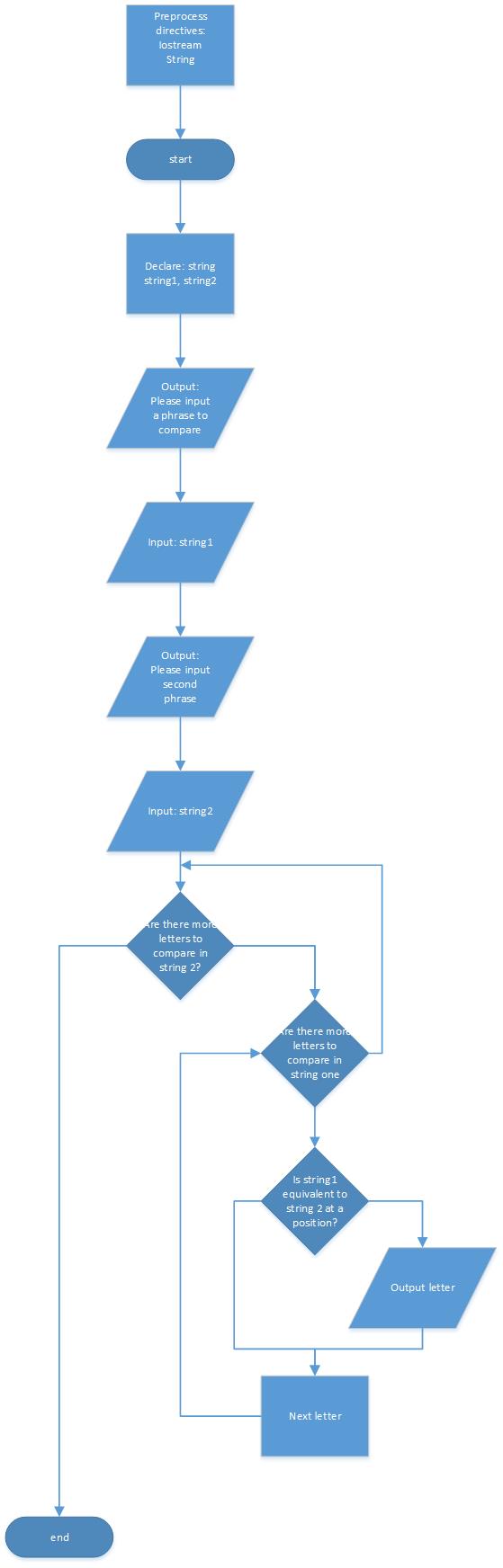
name.cpp flowchart



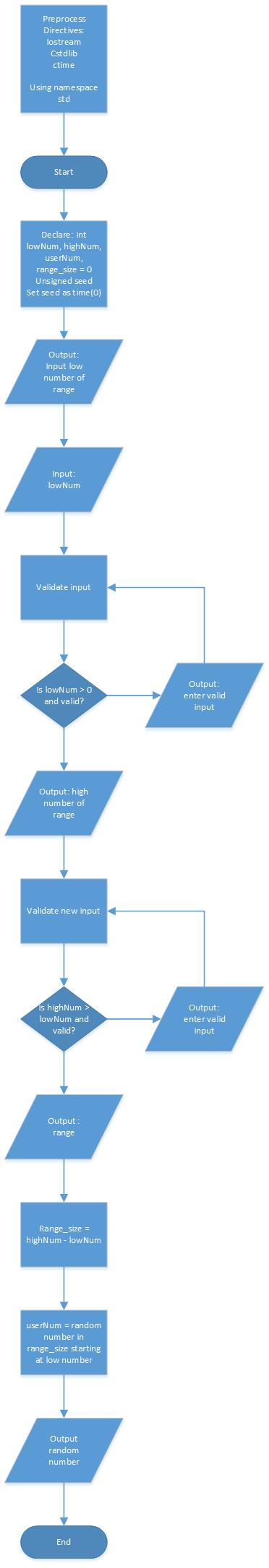
randNum.cpp flowchart



loop.cpp flowchart



randNum2.cpp flowchart



Word Guessing Game flowchart and design: See file: wordGuessDesign.pdf

Testing

strings.cpp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Word Entered | Expected Result | Actual Result | Phrase Entered | Expected Result | Actual | Num of Letters |
| chin | Word is spelled c,h,i,n Backwards  n,i,h,c | Expected | fish sticks | f,i,s,h, ,s,t,i,c,k,s  Backwards  s,k,c,i,t,s, ,h,s,i,f | Expected | 10 |
| 020 | 020 B: 020 | Expected | 0 0 0 | 0 0 0 B: 0 0 0 | Expected | 0 |
| df5 | d5f B:f5d | Expected | 00d 66g | 00d 66g B: g66 d00 | Expected | 2 |
| #! | #! B:!# | Expected | # yeah | # yeah B: heay # | Expected | 4 |

name.cpp

|  |  |  |
| --- | --- | --- |
| Name Entered | Expected Result | Actual Result |
| W L George | George, W L. | Expected |
| Will George | George, Will | Expected |
| Will | Please try again | Expected |
| 0032 531 | 531, 0032 | Expected |
| John Jacob Jingle-Heimer | Jingle-Heimer, John J | Expected |
| The big white whale | Please try again | Expected |

randNum.cpp

|  |  |  |
| --- | --- | --- |
| Input | Output Expected | Actual Output |
| None | 10 random numbers between 1 and 1000 | Expected |

loop.cpp

|  |  |  |  |
| --- | --- | --- | --- |
| Phrase 1 | Phrase 2 | Expected result | Actual result |
| Food | Food | Food | Fooood\* see notes |
| Fresh | Fish | Fsh | fsh |
| 1357 | 1658 | 15 | 15 |
| We go | Gone we | Goe we | Goe we |
| Good | opus | Oo | oo |

randNum2.cpp

|  |  |  |  |
| --- | --- | --- | --- |
| Num 1 Input | Num 2 Input | Expected Output | Actual Output |
| 5 | 10 | A number b/w 5 and 10 | 7 |
| 32000 | 35000 | A number b/w 32000 and 35000 | 34265 |
| -8 |  | That number is too low | Expected |
| 10 | 5 | Your max should be higher than your min | Expected |
| 32f5 |  | Quit program | Expected |
|  | 32f5 | Quit program | Expected |
|  | H | That is not an integer, try again | Expected |
| H |  | That is not an integer, try again | Expected |

Reflection

The purpose of this week was to introduce us to some of the functions and of using strings, for loops, random numbers and input validation. This was probably the toughest week so far but I learned a ton by getting started early and working with others in the class. As I result I’m far more confident in my abilities and understanding as we expand with our use of this language.

The strings program was helpful in understanding position and the length function of a string input. It was also valuable in understanding the cin and getline inputs as well as the isalpha input.

The name program was probably the most challenging of all the programs we made this week. As I started I thought it wouldn’t be too bad but I couldn’t figure out how to separate the words or account for someone entering 2 names instead of three. Eventually while working with my peers I was able to figure out that we needed use the find function to search the string for the desired character. Which, in this case was the space in between the words. Also there are many means of which someone can parse the words and so it took some exploring to figure out how to do it for this program. The best part about the more challenging programs is the satisfaction you get when the program works.

The first randNum program was very simple and took less than 10 minutes to plan and code. I believe the purpose was just to understand how the random number system works. I believe it will be a valuable tool to have moving forward. Also it was good for practicing the ‘for’ loops.

The loop program was challenging in the logic of the loops and what the program needed to achieve. To compare the strings you would need some nested for loops to look at each individual letter. This will be valuable for the word guessing program we will be designing for the project this week. One problem I did have with this program was with repeating letters next to each other when both words had them. I haven’t yet figured out how to prevent this but it’s something I will explore this coming week.

The second randNum program was focused on input validation more than anything. The thing I struggled with here was that I was unsure of what to look for. While I knew I didn’t want the user entering a letter instead of a number I didn’t think of what would happen if a user entered both an integer and a number until another user mentioned they were having problems with it. It took me a while to figure out what to do in that case, but I settled on it being a fatal error and just exiting completely out of the program.

Finally the word guessing game design was a fun project and I’m excited to start working on the code. It seems like one of the first programs where everyone’s result will come out a bit differently to come in line with the desired deliverables. I’m excited to have the opportunity to play around with it and see what we can do.