Anisha Aggarwal

CS 161

Homework #2

1. **“Guess My Number” – Algorithm**

1. Major steps
2. One player chooses a number from 1-100
3. Player 2 makes a guess
4. Player one will tell them if the guess is too high, too low, or just right
5. Repeat steps until guess is correct
6. Steps
   1. One player chooses a number from 1-100
   2. Player 2 guesses a number between 1-100
   3. Player 1 will tell them if the guess is too high, too low, or just right
      1. If guess is equal to the chosen number, player 1 will say “CORRECT!”
      2. If guess is lower then the chosen number, player 1 will say “too low”
      3. If guess is higher than the chosen number, player 1 will say “too high”
   4. After receiving input on guess, player 2 make another guess
      1. if guess was too low, make a guess between low guess and 100
         1. if guess is equal to the chosen number, player 1 will say “CORRECT”
         2. if guess is lower than the chosen number, player 1 will say “too low”
         3. if guess is higher than the chosen number, player 1 will say “too high”
         4. After receiving input on guess, player 2 make another guess
            1. if guess is lower than the chosen number, player 1 will say “too low”

player 2 will again make a guess but now between new low and 100

* + - * 1. if guess is higher than the chosen number, player 1 will say “too high”

player 2 will again make a guess but now between low and high guess

* + - * 1. Repeat step d. until player 2 makes a correct guess
    1. if guess was too high, make a guess between 1 and high guess
       1. if guess is equal to the chosen number, player 1 will say “CORRECT”
       2. if guess is lower than the chosen number, player 1 will say “too low”
       3. if guess is higher than the chosen number, player 1 will say “too high”
       4. After receiving input on guess, player 2 make another guess
          1. if guess is lower than the chosen number, player 1 will say “too low”

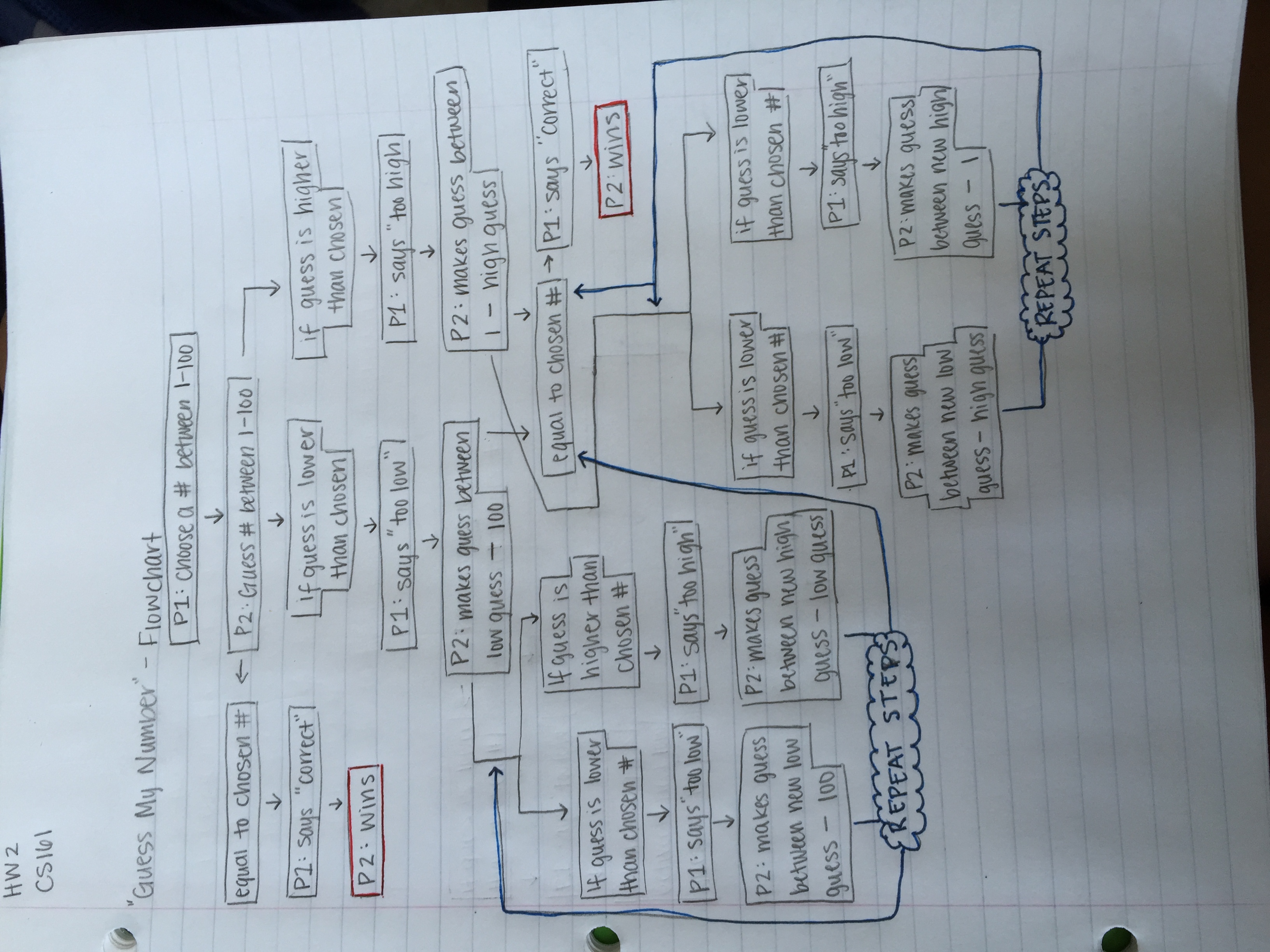
player 2 will again make a guess but now between low and high guess

* + - * 1. if guess is higher than the chosen number, player 1 will say “too high”

player 2 will again make a guess but now between 1 and new high guess

* + - * 1. Repeat step d. until player 2 makes a correct guess

1. **Flowchart**



2. SLiK program to C++ comparison

a. In order to crease a variable, in both cases the variable must be declared before the variable name can be used. Another similarity they have is that to create an integer variable before the name of the variable you say “int”. A difference is that to create a float variable, in SLiK you have to say “num” where as in C++ you say “float”.

b. A difference is that in C++, the prompts used before is “cout” with “<<” where as in SLiK the prompts used are “OUTPUT”. Another difference is that in C++ at the end of a cout line there must be a “endl” to start a new line where as in SLiK it automatically starts on a new line. A similarity is that they both are saying “out” to determine whether the data being read is going out.

c. A similarity between C++ and SLiK is that in C++ to read in the input they both have a clear way to say read “in” this information (C++ says “cin” and SLiK says “INPUT”). A difference between the two is that in SLiK to read in a line of data (with spaces), it must say “INPUT LINE” where as in C++ there is a “cin” prior to the line.

d. The calculations seem to be exactly the same. I don’t see any difference between C++ and SLiK. They both use the order of operations rules and the same symbols to mean add, subtract, etc.

3. **Review Questions (1, 2, 17-20)**

1. What does a declaration provide a variable?

b. data type

2. A variable’s data type describes all of the following except

d. the scope of the variable

17. What are non-executing statements that programmers place within code to explain program statements in English?

a. comment

18. Program comments are a form of internal documentation.

d. none of the above

19. Which of the following is valid advice for naming variables?

c. to make names easier to read, separate long names by using underscores or capitalization for each new word.

20. A message that asks a user for input is a prompt.

**Programming Exercises(1, 2)**

1. Explain why each of the following names does or doesn’t seem like a good variable name to you.
   1. d
      1. bad variable because there is no way to know what the variable is for
   2. dsctamt
      1. bad variable because it is hard to understand what the variable is for and it will be easy to misspell
   3. discountAmount
      1. good variable because is clear what the variable is for
   4. discount
      1. good variable because it is easy to know what it is for
   5. discoutAmountForEachNewCustomer
      1. good variable because the variable is clear what it is for
   6. discountYear2015
      1. good variable because it is clear what is for
   7. 2015Discountyear
      1. bad variable because it can’t start with numbers
2. If productCost and productPrice are numeric variables, and productName is a string variable, which of the following statements are valid assignments? If a statement is not valid, why not?
   1. productCost = 100
      1. valid assignment
   2. productPrice = productCost
      1. valid assignment
   3. productPrice = productName
      1. invalid – a sting variable cannot be assigned to a numerical variable
   4. productPrice = “24.95”
      1. invalid – product price is a numeric variable and is being string variable
   5. 15.76 = productCost
      1. invalid – the variable name must be on the left side
   6. productCost = $1,345.52
      1. invalid – dollar signs can’t be used and neither can commas
   7. productCost = productPrice – 10
      1. valid assignment
   8. productName = “mouse pad”
      1. valid assignment
   9. productCost +20 = productPrice
      1. invalid – variable that is being assigned a value should be on left side
   10. productName = 3-inch nails
       1. invalid – must have quotes on either side of the string
   11. productName = 43
       1. invalid – must have quotes around the string because it is not a numerical variable
   12. productName = “44”
       1. valid assignment
   13. “99” = productName
       1. invalid – variable that is being assigned must be on the left
   14. productName = brush
       1. invalid – must have quotes around the string
   15. battery = productName
       1. invalid - variable that is being assigned must be on the left
   16. productPrice = productPrice
       1. valid assignment – but it does nothing
   17. productName = productCost
       1. invalid assignment – you cant assign a number value to a string