CS 201 Homework 01
User Input And Output

Sarah Carter

September 14, 2019

Source Code Link: `https://github.com/SarahFDAK/CS201.git`
This homework took approximately *5* hours to complete.

1 Design

I had to do some research to get my brain in gear on the shape program, specifically, but once I'd stared at examples long enough, it started to make sense. I spent a lot of time plugging in numbers in my head to see if the numbers of spaces and pound signs would be correct.

2 Post Mortem

I initially forgot to include the error if the user inputs something other than a positive number, and I had to do some searching to find out how to make it a loop that could successfully accept new inputs. I learned a fair amount about cin.ignore() and cin.clear() and how useful they are for this type of situation. I created a function when I did the "greatest" program, and that was exceedingly educational and useful for more concisely entering repeated requests for input.

3 Answers to Questions

- The compiler translates source code as written by the programmer into object code that the computer can understand. When you run your program, the compiler translates the source code to object code, then the linker connects any object code from outside libraries to the user's object code and executes the program.
- A header is a file that contains definitions of terms used within the program. A compile-time error is an error found in the source code by the compiler. A linker connects the program's object code to the object code found in outside libraries that are defined in the include statement at the beginning of the program. A statement is the code within the program that specifically tells the computer what the program is supposed to do.
- A source file is the code written by the programmer that is most easily understood by users/people. An object file is the file created from the source code by the compiler that the computer can read and understand.
- Practicing concepts that you understand in theory from reading the text book helps to cement them in your mind, and also shows potential errors and variations that aren't covered in the examples in the book.
- A 'prompt' does just that - it prompts the user for some kind of input for the program to use in its output and/or calculations.
- backslash-n is a line ender that serves to end the line of output.
- An object is some memory saved to hold a specific type of value. A variable is a named object. A literal the literal notation of a value (i.e. assigning the literal "Hello, world" to a string variable).

- Literals can be assigned to strings, integers, and doubles, to name a few examples.
- Five names that are reserved and shouldn't be used are int, double, string, bool, and char. These are used to define variable types and making them actual variable names would be quite confusing.
- A conversion from a double to an int can be an unsafe conversion because doubles are allotted more memory than integers, so if a double has any decimals, they will be discarded when it is converted to an integer.

4 Sample Output

Listing 1: Sample Program Output

```
Please enter an integer to build a diamond pattern:
7
       #
      ###
     #####
    #######
   #########
  ###########
#############
  ###########
   #########
    #######
     #####
      ###
       #
```

5 My Program

```
1 #include <iostream>
2
3 int main(int argc, const char * argv[]) {
```

```
4
5       int user_num = 0; //Initialize and assign a variable to receive user input
6       std::cout << "Please enter an integer to build a diamond pattern: \n";
7       std::cin >> user_num;
8
9       for(int i = 0; i < user_num; i++) //Print the top of the diamond
10      {
11          for(int j = 0; j < (user_num - i - 1); j++) //Print spaces
12          {
13              std::cout << " ";
14          }
15          for(int k = 0; k < (2 * i +1); k++) //Print #s
16          {
17              std::cout << "#";
18          }
19          std::cout << std::endl;
20      }
21
22      for(int i = 0; i < user_num - 1; i++) //Print bottom of diamond
23      {
24          for(int j = 0; j < i + 1; j++) //Print spaces
25          {
26              std::cout << " ";
27          }
28          for(int k = 0; k < 2 * (user_num - 1 - i) - 1; k++) //Print #s
29          {
30              std::cout << "#";
31          }
32          std::cout << std::endl;
33      }
34      return 0;
35  }
```