Rec01

Note that everything you need to know about C++ for this recitation is contained in the document http://cis.poly.edu/jsterling/cs2124/LectureNotes/01.Intro.html.

You will write a single program for this recitation Clearly indicate with comments which parts of your program are for each of the steps below. Also whenever you are asked to output or display something, make sure that it is easy to tell *from your output* what it was that you are displaying. For example, if the instructions say “display x”, you might write a line of code such as:

```
cout << "x: " << x << endl;
```

Output

1. Write a hello … program without using namespace or any return statement.
    - Do *not* provide a “using namespace” directive.
    - You are going to write your output to the “console” otherwise known as “standard output”.
    - Build and run it.
    - What includes were needed?
    - What type does main return? Did you return anything? (no, not if you followed the instructions). For any function but main, this would be a problem. We’ll see this later.

2. Modify the program by adding a using namespace directive.
    - How can that simplify the code in main?
    - Write a second simpler line of code to print the output from the first activity. (Of course keep the original.) Your program should now print the same thing twice, but using slightly different syntax.

Static typing and defining variables

3. Define an int variable x without initializing it. Display it. What’s there? Don’t trust it!
    - Some things are guaranteed in the language and some things are ***undefined***. No matter what value you saw for x this time, the only valid answer is that the value is undefined. If you had compiled with some other compiler, you might have seen a different answer.

4. Define an int variable y initialized to 17 and an int z initialized to 2000000017, and finally a double pie with the value 3.14159.

- The value you are using for z is a little more than 2 billion.
- Which takes up more space in memory? How can we find out?
  - Display the sizeof(x), sizeof(y), sizeof(z) and sizeof(pie).

5. What do you expect to happen if you attempt to assign the string “felix” to x?
   - Yes, it will be an error. But what *kind*? Compilation or runtime?
   - Try it. Once you have tried it, *comment* it out.

Conditions

6. Check whether y is between 10 and 20, inclusive and display an appropriate message for either result.
   - In Python you have a nice concise notation for that
   - In most other languages, including C++, you have to break it into two tests, i.e. is y greater than or equal to 10 and is y less than or equal to 20.
   - Note that C++ does not use keywords, such as **`and`**, but instead uses special symbols **`&&`** for **`and`**, **`||`** for **`or`**, and **`!`** for **`not`**. Please note that even if your compiler does allow the use of those words, please, please, please, use the symbols.
   - In C++ you must put the test in parentheses. Python does not require them.

Looping

7. Print a line with the values from 10 to 20, inclusive.
   - Do it first with a for loop.
   - And then with a while loop.
   - And finally, just for the heck of it, with a do-while loop

8. Write a loop to repeatedly ask for a file name and attempt to open the file for reading until you succeed.
   - If you already have have an input file stream variable called `ifs` you can use it to open a file called “foo.txt” by calling `ifs.open(“foo.txt”)`.
   - What sort of loop is most appropriate for this.

9. After the file has successfully opened, read it word by word (i.e. whitespace delimited token) displaying each word on a separate line of output. Remember to close the file when you are done.

Vector

10. Define a vector to hold ints. Next, fill the vector with even integers from 10 to 100 inclusive.

11. Loop through the above vector to print out the values,
    - first using indices
    - then using a ranged for
    - And finally, in reverse!
        - Note, do not modify the original vector and do not create a new vector.
        - Also remember to use a size_t for your looping index

12. Define a vector holding the primes less than 20. You will fill this vector with the same line of code that defines it. (An example of doing this is in the slides posted on NYU Classes.) Display the vector any way you like, to make sure it has been properly initialized.

Functions

13. Write and test a function to display the values in a vector of ints, all values on one line.

14. Write (and test) a *function* to double the values in a vector of ints, using a traditional for loop to iterate over the vector. (You can use whatever vector you like.)
    - Needless to say, again

15. Again write (and test) a function to double the values in a vector of ints, however this time using a ranged for loop to iterate over the vector.
    - To actually change the values our loop variable can't be making copies of the entries in the vector.
    - Instead it needs to be a *reference*.