CSCI 1300

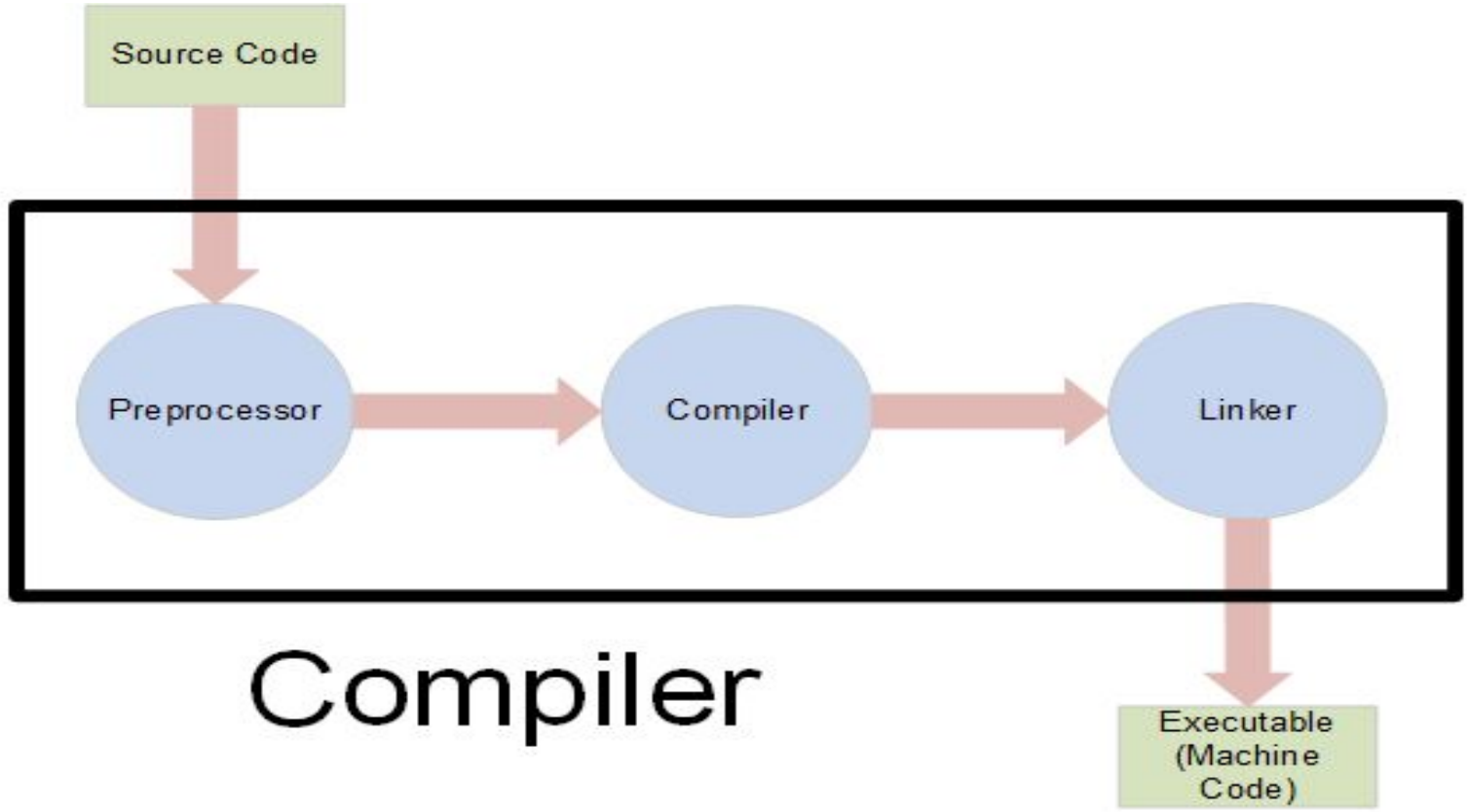
Introduction to C++
June 9th, 2021

Outline

- Overview of a compiler
- How to write C++ programs
  - Key components required and their meaning
  - Input output statements
  - variables and their types
  - IF/ELSE statements
  - loops (more specifically FOR loops and WHILE loops)

Please use the github link for the programing examples and slides.
https://github.com/rahul-aedula95/CSCI-1300

Compilation flow of a C++ program



Compilation flow of a C++ program

- Source code - This is the C++ program file written by you. They have file extension of (.cpp) which stands for c++.
- Preprocessor - Removes all the preprocessor directives (which are just statements referring to other snippets of code which get used in the program) and returns a c++ file.
- Compiler - Goes through line by line to check for errors and converts the c++ you wrote into machine code.
- Linker - creates a single executable file that can be used to run.

Note: This is an advanced subject in itself, however I wanted you to know the gist of it.

Key components of C++

- Preprocessor directives - gives instruction to the preprocessor perform a certain task.
  - Most common one we use is #include which tells the preprocessor to include snippets of code from somewhere else.
  - #define is a also a common one which can be used to replace values inside variables (we will talk about this at a later point)
- Namespaces - used to avoid variables with same name. Creates a profile as such. We will primarily be using the namespace std.
- main function - The first function to be looked at when compilation of a file begins. (We will also talk more about functions at a later point)
- ; (semicolon) used to indicate the end of a statement (with a few exceptions)
- { (curly brackets are used to signify scope)

Output in a C++ program

- Let us to refer helloWorld.cpp program in our week 2 github page.
- Output:
  - The purpose of having outputs is so that we can send messages to the so called user of the program (we will refer to this as user).
  - In c++ the most common way of sending standard output to your unix terminal is to use the statement cout which is a part of the std namespace.
  - Syntax: cout << “ Message you want to send to terminal inside quotes” << endl;
  - endl here is end line

Variables in a c++ program

- Let us to refer example2.cpp and example 3.cpp program in our week 2 github page.
- Think of them as named memory locations which can hold value based on a certain type.
- C++ is an older language so what type of value a variable can hold should be explicitly mentioned.
- These are the few common types which are used, there are more which I haven't mentioned
  - int - integer numbers
  - char - characters like letters
  - float - numbers which have decimal
  - double - same as float but can store bigger numbers.
- We can perform numerous operations on them.

Input in a C++ program

- Let us to refer example4.cpp program in our week 2 github page
- Input:
  - The purpose of these statements is to receive information from the user to then act on it.
  - Syntax: cin>> variable;

IF/ELSE statements

- Let us to refer example5.cpp program in our week 2 github page
- The main purpose of these statements is to use conditions to make a decision

logic:

```
        if (condition is met){
            do something}
        else{
            do something else
}
```

Loops

- Let us to refer example6.cpp and example7 program in our week 2 github page

The main purpose of looping statements is to run a command as many times as you need without having to type it again and again.

While loop logic:

```
while(some condition is met){
    do something
}
```

Loops (cont)

for loop will loop over a statement based on a preliminary set of conditions.

for loop logic:

```
for(t=0; t<desiredT; t++)
{
        do something;
}
```