Chapter 1: Introduction to Computers and C++ Programming

Instructor: Mark Edmonds

edmonds_mark@smc.edu

What is a program?

- A program is a set of instructions for a computer to follow
- We will write *source code* that is *compiled* into instructions the computer can execute
- C++ is considered a high-level language that allows programmers to write English-like sentences to describe a program, rather than low-level machine (binary; 1's and 0's) instructions for the processor

Introduction to C++

- C++ is a *compiled* language, meaning an entire program is converted into machine instructions at once
    - This is in contrast to *interpreted* languages, where the program is converted to machine instructions as the program executes (line by line)
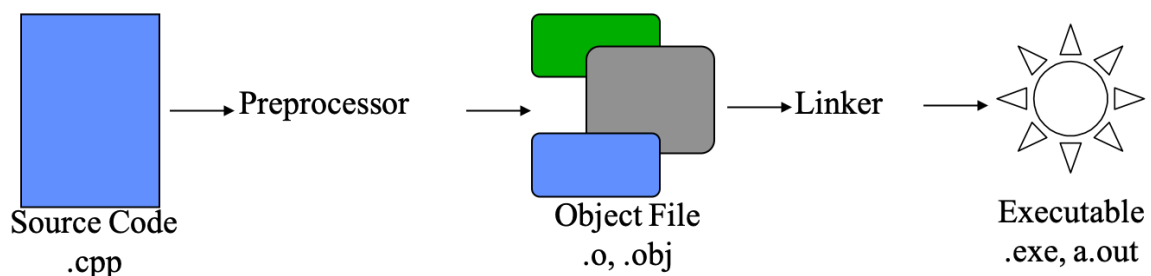

Figure 1: Compilation process

History of C++

- Authored by Bjorne Stroustrup, AT&T
- Extended the C Language
    - Added features to support Object-Oriented Programming
- C++ is considered a superset of C
- Language is now an international standard
    - American National Standards Institute
    - International Standards Organization

Program Development

- What is an *Algorithm*?
  - A sequence of precise instructions that leads to a solution
- Example algorithm: determine how many times a name occurs in a list of names, like [`Sarah`, `Bobby`, `Jack`, `Elizabeth`, `Sabrina`]. An algorithm may look like this:

```
1  Get the list of names
2  Get the name being checked
3  Set a counter to zero
4  For each name in the list:
5    Compare the name to the name being checked. If the names are the same
         , add one to the counter.
6  Return the value of the counter as the answer
```

- What is a *Program*?
  - An algorithm expressed in a language the computer can understand
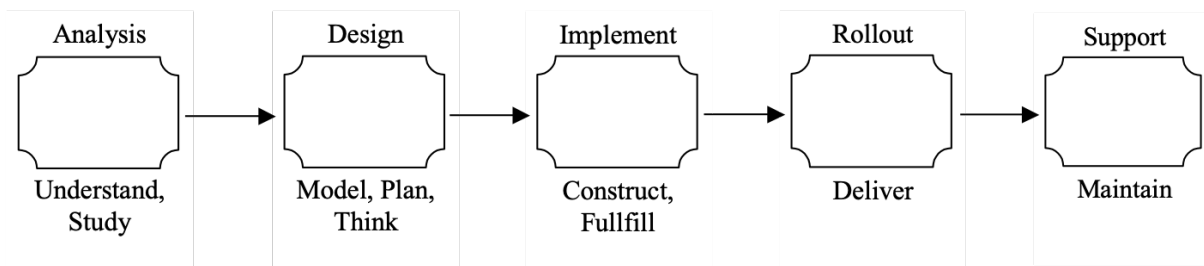

Figure 2: Program development process

Comments

- Before we look at our first C++ program, I wanted to talk about comments. Comments let you insert text into your program to describe what your program is doing.
- Comments in C++ can look like this:
  - `//double forward slash will comment to the end of the line`
  - `/*example of an enclosed comment. These comments can span multiple lines */`

Example hello_world.cpp

```
1  // This program displays a simple message
2  #include <iostream>  // for std::cout
```

```
 3  using namespace std; // supports the shorthanded cout instead of std::
       cout
 4
 5  // the main() function is the starting point of every program
 6  int main()
 7  {
 8    // print the content Hello, world! to the screen
 9    cout << "Hello World!\n";
10    // exit the program with no error
11    return 0;
12  }
```

Fixing Programs

- Most programs do not work initially
- There are multiple reasons why a program does not work:
    1. A *syntax error* means your source code cannot be converted into machine code. This source code cannot produce a valid C++ program, and therefore can never be run
    2. A *runtime error* means your program has executed into a state not permissible by the program or by C++; in this case your program will crash.
    3. A *logic error* means your program compiles and runs without error, but does not produce the expected result
        - Hardest error to fix!
- Fixing these problems consists of:
    1. *Syntax error*: examine the error and determine a fix in your source code
    2. *Runtime error*: examine the error and add error handling mechanisms to correct the error
    3. *Logic error*: examine your understanding the problem (can you solve it on a piece of paper by hand?) and make sure your program implements the same logic
- A *debugger* can be used to inspect your program as it executes
    - This helps fix runtime and logic errors
    - Allows you to trace the execution of a program and verify it behaves as you intend