Sam Graham

COP 1220-13815

8/30/2018

**Portability** – the ability to run on multiple systems without having to change much. Programing language C is consider a leader in portability. C compilers are available to many computer architectures. Programs that were designed to specifically interact with certain hardware devices or access special features of an operating system are usually not portable.

**Difference between source file, object file, and executable file**:

A source file is code that has been written in a text editor prior to going to a compiler. It’s bare code.

An object file contains source code that has been converted to machine language, but is not ready to run. It is missing startup code that interfaces between the program and the operating system. Object files are also missing the code for library routines (functions) that are part of the standard C library.

An executable file is the result of linkers merging correct startup code, C library code, and an object file. Executable files are ready-to-run.

**Seven major steps in programming:**

1. **Define the program objectives** - Think in general terms about what the program should do. What information it will need, how it will interact with that information, and what should it be returning?

2. **Design the program** – How will the program accomplish tasks outlined in the program objectives? Who are the target users and how will they interface with it? How should the program be organized? Decide on how data will be represented and which methods will process the data. Still think in general terms and not about specific code.

3. **Write the code** – Begin writing the code. You may sketch it out on paper, but at some point you’ll have to get it in the computer. Usually, writing code is done with a text editor to create a source code file. Be sure to document your work with comments.

4. **Compile** – Compile the code using your development environment of choice. Compilers convert the source code to executable code (machine language). Different computers have different machine languages and a C compiler translates into a particular machine language. Linkers bring libraries of standard routines like printf() and scanf(), to C compilers. The result is an executable file that a computer can understand and a user can run.

5. **Run the program** – Running an executable file common environments, like Windows Command-Prompt or Linux terminal mode, is as easy as typing the name of the executable file. Other environments may require a run command. Integrated Development Environments (IDEs) let users edit and execute programs within the IDE. The program can run from the operating system by clicking the filename or icon.

6. **Test and debug the program** – Check to be sure your program is doing what it is supposed to. Look for design errors, ensure ideas were implemented correctly, be sure that unexpected input will not mess things up, check your syntax, and look for logical errors.

7. **Maintain and modify the program** – The more a program is used, the more likely you are to find reasons to make changes to it. You may uncover hidden bugs that were overlooked, or decide that a new feature would enhance the value. Maintenance and modification are made easier with good documentation and following sound design practices.

**Pick my C Compiler. Which one was it?**

I plan to use Microsoft Visual Studio. I understand that I’ll have to choose the C++ option, then Win32 Console, then Empty Project in Application settings. If it becomes too much of a hassle, I will get a command-line compiler. I’m leaning toward MinGW.