

II. C/C++ Programming

II.1. C/C++ programming essentials

files

source file (.c, .cc, .cpp, .cxx)

Almost all computer programs must be written in text files. Even visual environments, in which many of the text files are generated automatically, still require you to work with the text files part of the time. Use a text editor to create this file. Do not use a word processor, as word processors insert extra formatting characters that cause problems later on. The .c extension is for C source code and the others are for C++ source code in different systems. Use the .cc extension for this course.

executable file (.exe)

Computer programs can be “compiled” or “interpreted”. In an interpreted program, another program, called the “interpreter” must be running to execute the source program. This program reads the source file, interprets the commands and then performs them. In compiled programs, a compiler reads the source file and then generates an executable program as a separate file that can then be run directly by the operating system. Our C/C++ programs will be compiled. The .exe extension is not required and we will not be using it in this course.

comments

The first few lines of the source file should contain comments about the code in that file, including:

- Project (or course) name,
- Organization name,
- Copyright notice,
- The name of the file,
- The name of the module/library that it is in,
- The name of libraries that are required,
- The names of the authors,
- Overview and general comments.

Comments should also be distributed throughout the code to explain WHY you are doing something. The comments should not explain WHAT you are doing, because this should be obvious from the code, if it is well written.

include section

After the comments, the next section of code contains the #include compiler directives. This directive causes the named file to be read in as if it were typed right there. The file “iostream” (/opt/SUNWspro/WS6U2/include/CC/Cstd/iostream) includes information needed to compile cout below.

the main() function

The program is written as a single function (or subroutine). Every C++ program must have a main() function. The main() function is special because it is the one that is called to start the program. Note that the main function returns an integer value. The function code is contained within the curly brackets.

defining the variables

Before any variables can be used, they must be defined, so that the compiler knows what type they are. For example, “int x;” defines the variable x to be an integer.

expressions/statements

Variables are assigned new values based results of calculations.

Other statements can be used to control the flow of the program.

simple input/output (IO)

cin means “console in” and cout means “console out” (console is text terminal).

The << operator shows what is supposed to be input and printed out. It can be numbers, text, etc.

‘\t’ – tabs

endl – forces printing a new line.

An example program

```
/* MEE5408 Design Automation: Theory and Implementation
```

```
   Mechanical Engineering-Engineering Mechanics Dept.  
   Michigan Tech University  
   Copyright (C) 2001. All Rights Reserved.
```

```
FILE:                add.cc  
MODULE:              none  
DEPENDS ON MODULE(S):  none  
AUTHOR(S):           Bernie Bettig
```

```
OVERVIEW  
=====
```

Code for adding two numbers.

```
*/  
  
#include <iostream>  
  
int main()  
{  
    int a, b, sum;  
    cin << a << b;  
    sum = a+b;  
    cout << "sum = \t" << sum << endl;  
    return 0;  
}
```

Using the compiler

Bring up a UNIX terminal window. Type:

```
>CC assignment1.cc
```

This will result in an executable file called `a.out`. Type:

```
>CC assignment1.cc -o assignment1
```

This will result in an executable file called `assignment1`. If you have “compile-time” errors, you will need to make changes in the source file, re-save it, and re-compile. To let the operating system know where your program is type:

```
>setenv PATH .:$PATH
```

This will put the current directory (indicated by “.”) on the search path for executing commands. This can also be done in your UNIX startup scripts. (Talk to a system administrator.) To start the program type:

```
>assignment1
```

This will start the program. If you did not set the path, you should type “`./assignment1`” to execute the file, if it is in the current directory.

Using a development environment

Instead of using the command line to compile, you can use a windows program. Type:

```
>workshop &
```

or simply start up the Visual C++ environment on the PC's.