

Source File: ~/1337/05/lab05. (C|CPP|cpp|c++|cc|cxx|cp)
Input: None
Output: Screen
Value: 1

Use the program below as a starting point for Lab 05.

```
// Your name
// CS 1337
// Lab 05

#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, World!" << endl;
    return 0;
}
```

Read through and carry out the commands shown in Figure 1.

```
1 newuser@csunix ~> # Commands beginning with # are comments and do not need
2 newuser@csunix ~> # to be entered.
3 newuser@csunix ~> #
4 newuser@csunix ~> # Note that the command prompt (the information to the left of
5 newuser@csunix ~> # >) shows the current userid, the host name, and the
6 newuser@csunix ~> # current directory. The current directory is shown as
7 newuser@csunix ~> # ~, which is your home directory.
8 newuser@csunix ~> #
9 newuser@csunix ~> # It is recommended that you create a folder for
10 newuser@csunix ~> # organizing your assignments for this class.
11 newuser@csunix ~> # From the home directory, create a directory for 1337.
12 newuser@csunix ~> mkdir 1337
13 newuser@csunix ~> # Change to the directory created above.
14 newuser@csunix ~> cd 1337
15 newuser@csunix ~/1337> # Note how the prompt changes to reflect the new directory.
16 newuser@csunix ~/1337> # From the 1337 folder, create a directory for lab 05.
17 newuser@csunix ~/1337> mkdir 05
18 newuser@csunix ~/1337> # Change to the directory created above.
19 newuser@csunix ~/1337> cd 05
20 newuser@csunix ~/1337/05> # Copy the sample input and output files for this lab.
21 newuser@csunix ~/1337/05> # The Unix command for copy is cp. The next argument
22 newuser@csunix ~/1337/05> # to the cp command is the path where the file(s) will
23 newuser@csunix ~/1337/05> # be copied from. The wildcard character * matches all
24 newuser@csunix ~/1337/05> # files in the folder. The last argument to this
25 newuser@csunix ~/1337/05> # particular cp command is . (dot), which means to
```

Figure 1. Commands to Compile, Link, & Run Lab 05 (Part 1 of 3)

```
26 newuser@csunix ~/1337/05> # save the copied file(s) to the current directory
27 newuser@csunix ~/1337/05> # with the same name as in the first argument.
28 newuser@csunix ~/1337/05> cp /usr/local/1337/data/05/* .
29 newuser@csunix ~/1337/05> # The ls command lists the contents of a directory.
30 newuser@csunix ~/1337/05> # Note that the output from ls confirms that two
31 newuser@csunix ~/1337/05> # files were copied.
32 newuser@csunix ~/1337/05> ls
33 01.dat 01.out
34 newuser@csunix ~/1337/05> # Create an empty file that will contain the source
35 newuser@csunix ~/1337/05> # code for this lab.
36 newuser@csunix ~/1337/05> touch lab05.cpp
37 newuser@csunix ~/1337/05> ls
38 01.dat 01.out lab05.cpp
39 newuser@csunix ~/1337/05> # Using the -l option with the ls command lists the
40 newuser@csunix ~/1337/05> # contents of the directory in long format. The long
41 newuser@csunix ~/1337/05> # listing displays several attributes about files,
42 newuser@csunix ~/1337/05> # including permissions, the owner, the group, the
43 newuser@csunix ~/1337/05> # number of bytes the file contains, the date and
44 newuser@csunix ~/1337/05> # time the file was last modified, and the path
45 newuser@csunix ~/1337/05> # to the file. Note the output from the long listing
46 newuser@csunix ~/1337/05> # confirms that the cpp file contains zero bytes.
47 newuser@csunix ~/1337/05> ls -l
48 total 4
49 -rw-r--r-- 1 newuser students 0 Jul 11 08:29 01.dat
50 -rw-r--r-- 1 newuser students 60 Jul 11 08:29 01.out
51 -rw-r--r-- 1 newuser students 0 Jul 11 08:29 lab05.cpp
52 newuser@csunix ~/1337/05> # Edit lab05.cpp using Notepad++ and upload it to
53 newuser@csunix ~/1337/05> # the server. Doing another long listing confirms
54 newuser@csunix ~/1337/05> # that lab05.cpp has increased in size and the
55 newuser@csunix ~/1337/05> # timestamp information has changed.
56 newuser@csunix ~/1337/05> ls -l
57 total 5
58 -rw-r--r-- 1 newuser students 0 Jul 11 08:29 01.dat
59 -rw-r--r-- 1 newuser students 60 Jul 11 08:29 01.out
60 -rw-r--r-- 1 newuser students 140 Jul 11 08:32 lab05.cpp
61 newuser@csunix ~/1337/05> # Invoke the C++ compiler using the g++ command.
62 newuser@csunix ~/1337/05> # The -c option means to compile only.
63 newuser@csunix ~/1337/05> # The -Wall option means to show all warnings.
64 newuser@csunix ~/1337/05> # The -std=c++11 option means to use the C++11 standard.
65 newuser@csunix ~/1337/05> # The -g option means to include symbolic
66 newuser@csunix ~/1337/05> # information in the object code so that a
67 newuser@csunix ~/1337/05> # debugger can be used.
68 newuser@csunix ~/1337/05> g++ -g -Wall -std=c++11 -c lab05.cpp
69 newuser@csunix ~/1337/05> # If the source code is syntax error free, an object
70 newuser@csunix ~/1337/05> # code file will be created.
71 newuser@csunix ~/1337/05> ls -l
72 total 119
73 -rw-r--r-- 1 newuser students 0 Jul 11 08:29 01.dat
74 -rw-r--r-- 1 newuser students 60 Jul 11 08:29 01.out
```

Figure 1. Commands to Compile, Link, & Run Lab 05 (Part 2 of 3)

```
75 -rw-r--r-- 1 newuser students 140 Jul 11 08:32 lab05.cpp
76 -rw-r--r-- 1 newuser students 57624 Jul 11 08:33 lab05.o
77 newuser@csunix ~/1337/05> # Invoke the linkage editor using the g++ command
78 newuser@csunix ~/1337/05> # once again. The input to the linker is the object
79 newuser@csunix ~/1337/05> # code file created from the compilation step. The
80 newuser@csunix ~/1337/05> # output (-o option) from the linker is an executable
81 newuser@csunix ~/1337/05> # file.
82 newuser@csunix ~/1337/05> g++ -o lab05 lab05.o
83 newuser@csunix ~/1337/05> # If there are no link errors, an executable file
84 newuser@csunix ~/1337/05> # will be created.
85 newuser@csunix ~/1337/05> ls -l
86 total 120
87 -rw-r--r-- 1 newuser students 0 Jul 11 08:29 01.dat
88 -rw-r--r-- 1 newuser students 60 Jul 11 08:29 01.out
89 -rwxr-xr-x 1 newuser students 55084 Jul 11 08:34 lab05
90 -rw-r--r-- 1 newuser students 140 Jul 11 08:32 lab05.cpp
91 -rw-r--r-- 1 newuser students 57624 Jul 11 08:33 lab05.o
92 newuser@csunix ~/1337/05> # To execute the program, type the name of the
93 newuser@csunix ~/1337/05> # executable at the prompt.
94 newuser@csunix ~/1337/05> ./lab05
95 Hello, World!
96 newuser@csunix ~/1337/05>
```

Figure 1. Commands to Compile, Link, & Run Lab 05 (Part 3 of 3)

Before submitting your assignment, modify the source program to produce the additional lines of output as shown in Figure 2.

```
1 newuser@csunix ~/1337/05> g++ -g -Wall -std=c++11 -c lab05.cpp
2 newuser@csunix ~/1337/05> g++ -o lab05 lab05.o
3 newuser@csunix ~/1337/05> ./lab05
4 Your Name - CS 1337 - Lab 05
5
6 Hello, World!
7
8 Goodbye, Moon.
9
10 newuser@csunix ~/1337/05> # Now execute the program once again and redirect
11 newuser@csunix ~/1337/05> # the output from the standard output device to
12 newuser@csunix ~/1337/05> # an ordinary file (note the > in the below command).
13 newuser@csunix ~/1337/05> ./lab05 > my.out
14 newuser@csunix ~/1337/05> # Now see if there are differences between the .out
15 newuser@csunix ~/1337/05> # files. Recall that 01.out was copied from the folder
16 newuser@csunix ~/1337/05> # /usr/local/1337/data/05/ and contains the expected
17 newuser@csunix ~/1337/05> # output. The file my.out contains the output produced
18 newuser@csunix ~/1337/05> # by your program. A difference in the ID line is
19 newuser@csunix ~/1337/05> # acceptable.
20 newuser@csunix ~/1337/05> diff 01.out my.out
21 newuser@csunix ~/1337/05>
```

Figure 2. Commands to Compile, Link, & Run Lab 05

When your program has been debugged, you are ready to submit it for credit. Use Blackboard to submit your source code.