## CISC105 Fall 2006 Lab01

- Review the code examples from class.
- Write a program for each of the following problems. If you wish, you may start each program using a previous program as a base and then modifying it, BUT you will learn more if you code each one from scratch. Be sure to save every separate program as its own .c file. All programs must be properly commented and indented (see Assignment Standards on the class website).
- Some programs below are associated with a question. Answer the questions using C comments below your code in the program file.
- Use the C arithmetic operators +, -, \*, and /.
- During your scripting of the programs, you will compile and run each one. If a program does not compile, don't try to run it (because a.out will not have been created). Only one file below will not compile, but several have errors when they run. You will not be able to run the file that does not compile.

## **Programs**

For each numbered problem below you will write a small program (except the last one). Name each program lab01.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab01.1.c

- 1. Declare an integer variable named i, assign it a value of 23, and print a message and the value as we did in class. Remember that you need a format specifier (also called a "placeholder"; see sections 2.1 and 2.2 of your Hanly & Koffman text).
- 2. Declare an integer variable named "ratCount" and repeat the previous program.
- 3. Copy the first program using the shell command "cp" (what will the copy name be?). In the new copy, change the name of the variable i to i21. Be sure to change the name everywhere the variable is used. Compile and run to show it works (be sure to do this in your script!). Now edit the file again and change the name to 21i. What error message do you get when you compile?
- 4. Write a new program in which you declare three integer variables (i, j, and result) and assign them values to calculate and display

 $result = i^2 + 11j - 10$  (this is not C code)

for the case where i = 73 and j = 15. Use only the arithmetic operators from 2.5 of your textbook! Your code will not look exactly like this formula - think, and test your ideas.

- 5. Follow these instructions carefully and do not add steps. Copy the previous program into a new file. Change the value of i to 2 and the value of j to 7. Show that it works. Verify your results. Then edit the program as follows: After you print the result, add *only* an assignment statement to change either one of the variables (i or j) and then a statement to print result again. Did result change when you changed i or j? Explain why or why not.
- 6. Copy program 1 using the shell command "cp". In the new copy, change the value you assign to i to 98.6 (which is not an integer). What does the program print? Why?

- 7. Declare a variable named x of type "double" instead of integer. Assign it the value 2.123. Print it out with a nice message, but change the format specifier from %d to %lf. You should see the complete number when you compile and run. Now change to the format specifier for an integer, %d. What gets printed when you compile and run? Explain why in your comments. After you test this, change your program back so that it prints correctly.
- 8. Declare an integer variable i, assign it the value 7, and print the integer result of dividing i by the integer 2. Re-read the text if you don't know how to divide in C.
- 9. Declare a double variable f, assign it the value 7, and print the result of dividing f by 2.
- 10. Review the Unix shell commands ls, cd, rm, mv, and mkdir. Practice making one or more directories, moving into them, listing the files and directories there, and then moving back out. Use rm to remove any practice directories you have made. Then create a directory "cisc105". Move into it and create directories "lab00" and "lab01". Then move your lab files from your home directory (where you are if you just type "cd" with no arguments and hit return) to the appropriate lab directories using mv. When you script your work, show that you can move into and out of these two directories using cd, and list (show) the directory contents of each.

You should have a total of 9 programs named lab01.1.c to lab01.9.c. Make a single script file (see lab00 for the scripting instructions) where you cat, compile, and run each one in its final form (if it didn't compile, don't run it in the script - mark the place in the printed script file with a colored marker so it stands out). After all files have been run in the script, use ls and cd to show your new directories and their files in the script.

## On the first page of every printed copy for this course, your name, section, and TA's name must appear.

Submit all program and script files on MyCourses before midnight Thursday of next week, and give the paper version to your TA at the beginning of your Friday lab (or in lecture Friday if you have a Wednesday lab). Note: cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.