CISC105 Fall 2006 Lab02

- 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.
- Note that I am assuming you remember what you did in previous labs. For example, you are now expected to know if you need to declare a variable for a certain program. If you have any questions, be sure to ask your TA that is the whole purpose of a lab period!
- Remember that comments are an important part of your program. Every program must have your **name, TA, section,** and a **description** of what the program does, and every function must have its own comments. Programs that have complex innards need comments sprinkled throughout.
- Any time you expect user input, you must print a **prompt** telling the user what to enter.

Programs

For each numbered problem below you will write a small program (except the last one). Name each program lab02.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab02.1.c

- 1. This program will compare two numbers and print a message. Have the program test if 2 is less than three, and print "Two is less than three" if it is. Don't add other features.
- 2. Use a shell command to copy the previous program, and add the feature that it prints "Two is not less than three" if that is the case.
- 3. Use scanf to get two integers from user input, and then print the integers with a nice message. Can you do this on paper without looking at your notes? Did you remember the prompt? Experiment with using two separate scanfs or a single scanf to do the work (your text has lots of examples).
- 4. Use a shell command to copy program 2. Add scanfs (and prompts!) so the two integers that are compared are entered by the user. Be sure to change the print statements so that the user's numbers are printed by the messages.
- 5. In class we discussed three things that we put in a program so that we can have our own functions. What are the three things? Write a program with two functions: a main(), and a new function named sayHello. The sayHello function will have an empty parameter list, and the only code inside it will print the message "Hello!". Have your main() function call your new function so that the message prints. There should not be any printing done directly from main (except perhaps a newline).
- 6. Write a program with a function (give it a meaningful name) that takes as parameters one integer and one double. The function should print the sum of the two numbers. (What format specifier will you need to use?)

- 7. Write a program that gets two integers from a user (placing them in integer variables) and prints a nice message showing the result of integer division and double division. (How will you get double division to happen if both variables are integers? Hint: see table 2.9 in your text).
- 8. Write a program with a function (give it a meaningful name) that takes as parameters two integers and prints the larger integer with a meaningful message. In main, use scanf to get two integers from the user and pass them to your new function. Hint: test the function by calling it from main on two integer constants after you know it works correctly, add the scanfs.

You should have a total of 7 programs named lab02.1.c to lab02.7.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.