

CSE 105, Lab 1, Summer 2006

Instructor: Cory Strobe

July 11, 2006

1. Setup for Lab 1.

- Create a new directory for Lab 1.
 - From the root directory change to the class directory: `cd cse105.`
 - Create a new subdirectory for Lab 1: `mkdir Lab1.`
- Download the files from the web pages:
 - `cse.unl.edu/~cstrobe/csce105su06/lab1/lab1.c`
 - `cse.unl.edu/~cstrobe/csce105su06/lab1/lab1_buggy.c`
 - `cse.unl.edu/~cstrobe/csce105su06/lab1/lab1a.c`
 - `cse.unl.edu/~cstrobe/csce105su06/lab1/lab1a.dat`
- Copy and paste each file into the Lab 1 subdirectory.

2. Compiling a source code program.

- Source code files: Text files ending in the extension `.c` containing code.
- Invoking the compiler: `cc sourceCodeFileName.c` generates an executable with the default name `a.out`.
- Running default executable `a.out`: Type `./a.out`.
- Changing the executable name: `cc -o newName sourceCodeFileName.c` generates an executable labeled `newName`, which is run by typing `./newName`.

3. Compiling lab1.c

- Type `cc lab1.c` and hit enter to generate `a.out`.

- Run `a.out`, resulting in the prompt:

Enter number of whole dollars => *enter 3*
 3 dollars can buy you 36 doughnuts!

4. Reading compiler errors

- Compile `lab1_buggy.c`. There are several bugs (errors in the way the code is written that result in compiler errors) in this code. As you attempt to correct each error, make note of both the compiler error message and the cause/solution.

1. First error message →

cause/correction →

2. Second error message →

cause/correction →

3. Third error message →

cause/correction →

5. **redirection**: Takes input from a file, and redirects output to a file.

> directs output from the program to the file named on the right hand side of the operator.

`./a.out > output.txt`

< feeds input from the file named on the right hand side of the operator to the program (**Batch mode**).

`./a.out < data1.txt`

Note that data1.txt must already be created in order to use this operator.

- Compile the code of `lab1a.c` to the executable named `convertMiles` and run it. Notice that it asks for data to be entered four times.
- Now type `convertMiles < data.txt > output.txt`. Open the file `output.txt` using `pico`.
- Briefly describe what you see in this file.

6. **Problem:** Chapter 2, problem 5:

- Write a program to take two numbers as input data and display their sum, their difference, their product, and their quotient.
 - Problem inputs: two items, `double x,y`
 - Problem outputs:


```
double sum
double difference
double product
double quotient
```
- Email your work to me before leaving lab today. `cstrope@cse.unl.edu`