This lecture talks about the first homework. This homework is the foundation for all programming assignments.

This assignment is primarily reading and understanding.

This assignment explains how programming assignments are structured and graded.

Go to the E C E two six four directory. There are two directories created earlier, called lectures and problems. The directory problems contains the files of the problems from the git hub repository.

Create a new directory called solutions to store the files for your solutions. The files in this directory will be uses for submitting homework assignments.

The C P minus R command copies the entire directory and all files inside the directory.

Enter the directory and there are quite a few files.

You can find the files in the git hub website as well. This is the U R L of the first assignment in the repository.

In an earlier lecture, I mentioned that there are three different file formats: text format, executable format, and object format. We can use G. C. C. to convert from the text format to the executable format or the object format.

Let’s study G. C. C. more.

G C. C. can accept many different options. In this class, we will use G. C. C. in this way:

Add minus S. T. D. equals C. 99, minus G., minus W. A. L. L., and so on.

These enable a set of functions in G. C. C.

.

The first one tells G. C. C. to use the C. standard announced in 1999.

The second one, minus G., enables debugging.

The next two enable warning messages.

It is understandable that typing this long command is tedious. A simple solution is to create an alias, in the same way we created an alias for al as earlier.

By creating an alias in dot B. A. S. H. R. C., these options for G. C. C. are always added when you type G. C. C. in the future.

Let’s see how G. C. C. ‘s warning can be our first-line of defense to detect mistakes.

Consider a program that has a variable that is not used. An unused variable is usually caused by mistyping.

G C. C. can detect the unused variable.

This situation is a little trickier. A variable called V. A. R. is created outside for. Another variable with the same name is also created inside for. This is allowed but should not be used. This is called a shadow variable.

Shadow variables are likely to become confusing because the value inside for can be different from the value outside for.

Again, G. C. C. can detect this situation and issue warning.

The next is called conditional compilation. By adding I. F. D. E. F. and E. N. D. I. F., it is possible to turn on and off code using G.C.C.

.

Suppose you want to print a message only during debugging, you can put enclose that message inside I. F. D. E. F. debug and E. N. D. I. F.

.

If you add minus D debug, this debugging message is printed. Minus D after G. C. C. means defining the symbol.

You can also use I. F. N. D. E. F. This message is printed if debug is not defined by G. C. C.

.

By defining or not defining specific symbols, you can have different sections of code turned on or off. This allows you to experiment different solutions.