

CSCI 2270

Data Structures and Algorithms

Elizabeth White
elizabeth.white@colorado.edu

Office hours: ECCS 128
Wednesday 1:00-2:00pm
Thursday 2:00-3:00pm

Data structures

- Specific data structure goals for the class:
 - Write and manipulate arrays, linked lists, trees, graphs
 - Generalize these to work with many data types
 - Know how to apply template classes
 - Learn to write safe, smart code for classes
- General data structure goals for the class:
 - Select proper structures for various problems
 - Understand how to design data structures that suit particular algorithms

Algorithms

- Specific algorithmic goals for the class:
 - Compare how different techniques for solving a problem perform as the problem size grows
 - Select correct pairings of algorithms & data structures
 - Measure and compare time for algorithms
 - Know when and why to apply recursion
- General algorithmic goals for the class:
 - Learn how to teach yourself new algorithms
 - Learn how to make an existing algorithm more efficient

General info, part 1

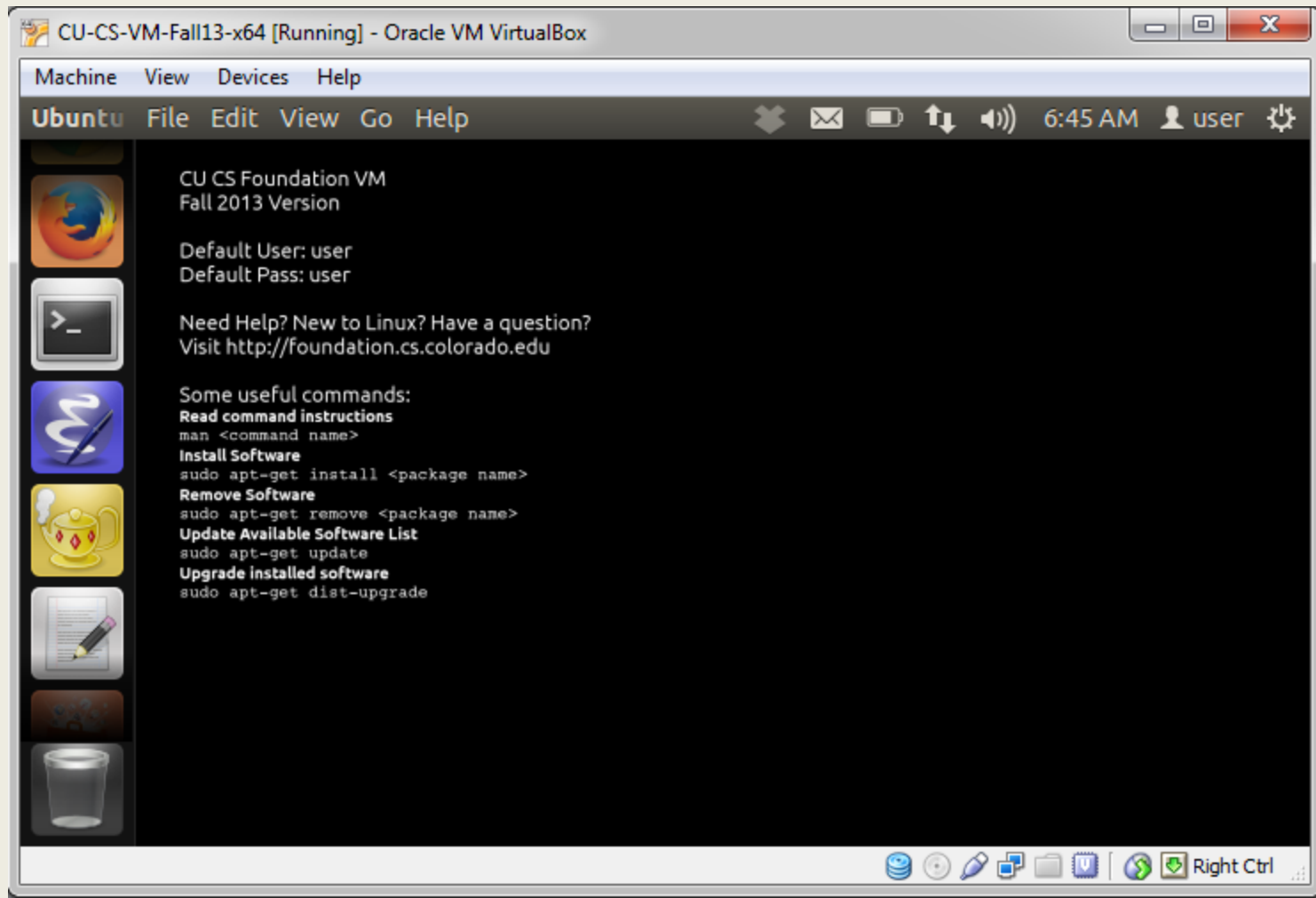
- Moodle web site: moodle.cs.colorado.edu
- Experts and novice programmers, mixed together
 - Don't be a jerk if you are an expert
 - Don't assume everyone else is an expert if you are a novice
- Small programming assignments for most lectures
- Bigger homeworks assigned about every 2 weeks
 - Interview grading (new to some of you)
 - Working together is fine, if you document it
 - You must still *type everything yourself*
 - You must do your best to *understand what you type*
- Research talk attendance and summary
- Recitations: mandatory

General info, part 2

- Textbook—Introduction to Algorithms, 3rd edition, by Cormen, Leiserson, Rivest, and Stein
 - Expensive, but I still use my copy a lot (try Google!)
- Additional C++ specific readings from online sources
- 3 midterms
 - Written component in class Friday, September 26, Friday, October 24, and Friday, November 21
 - Interview graded programming component during weeks of September 26, October 24, and November 21
- 1 final (in MATH 100):
 - Thursday December 18th, 7:30pm-10:00pm

The Virtual Machine

- <http://foundation.cs.colorado.edu/sde>
- Imitates the lab machine environment



Friendly Virtual Machine

- You'll need the fall 2014 version of the VM to stay current in the class. We'll have some flash drives for downloading in recitation this week.
- If you would like some in-person help getting your copy of the CU CS VM setup, please attend one of the following install sessions. All install sessions are held in the [CSEL Student Lab](#) (**ECCS 112**). We will provide USB flash drives with copies of the VM image for your convenience.
- 08/25/14 - 11:00 AM to Noon - VM Install Session
- 08/26/14 - 11:00 AM to Noon - VM Install Session
- 08/27/14 - 2:00 PM to 3:00 PM - VM Install Session
- 08/28/14 - 2:00 PM to 3:00 PM - VM Install Session

Hello World in C++ (hello_world_1.cpp)

```
// Including this library gives us the C++ classes and functions for printing to
the screen (cout, endl).
#include <iostream>

// The classes and functions for printing in iostream are in a namespace called
std.
using namespace std;

// the code always begins to run here, at the function called main
int main()
{
    // print the text to the screen (cout) and then a new line (endl)
    cout << "Hello World" << endl;
    // main needs to send back an integer;
    // under normal circumstances, this is 0
    return 0;
}
```

g++ -Wall -o hello_world_1 hello_world_1.cpp

Alternative Hello World in C++

(hello_world_2.cpp)

// Including this library gives us the C++ classes and functions for printing to the screen (cout, endl).

```
#include <iostream>
```

// We can take out the using namespace std; line below:

```
// using namespace std;
```

// the code always begins to run here, at the function called main

```
int main()
```

```
{
```

```
    // print the text to the screen (cout) and then a new line (endl)
```

```
    std::cout << "Hello World" << std::endl;
```

```
    // main needs to send back an integer;
```

```
    // under normal circumstances, this is 0
```

```
    return 0;
```

```
}
```

```
g++ -Wall -o hello_world_2 hello_world_2.cpp
```

What's this mean?

```
g++ -Wall -o hello_world hello_world.cpp
```

g++: a compiler for C++ code

-Wall: requests compiler to tell you if your code raises any warnings (you should avoid code that generates warnings)

-o hello_world: specifies the name of the executable file we get as a result of the compilation

hello_world.cpp: the file we're compiling

What about a less dumb example?

Lab this week: letter shifter (lowercase, no letters except a-z)

“abc” shifted by 3: “def”

“abc” shifted by 26: “abc” again

“abc” shifted by 25: “zab”

You’ll do this using 3 small functions:

print a string to the screen

read a string from the keyboard

compute a shifted string from an existing string

And you’ll write one bigger function that uses these little functions to get a string from the user, get a shift distance from the user, and print the shifted string.

Make sure you upload your files to the Moodle and show your TA when you get this working.