CSCI 1300 - Intro to Computer Programming

Instructor: Knox

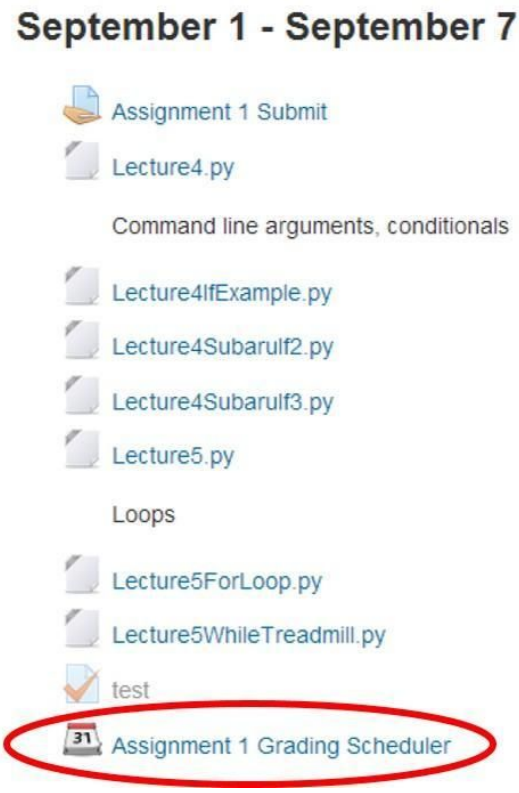
Recitation 3

There will be a section below that is due at the end of recitation.

## I. Scheduling a Grading Meeting on Moodle

- Reserve a time slot for assignment 1 grading in Moodle with your TA.
- Rooms for grading sessions will be based on your sign-up time. ( for example: ECCR 1B09/ 1B50/ 1B50E/ 1B50F or ECCS 102/ 112A)
- Please attend the grading meeting on time and keep your committed time. There are no extra meeting slots if you miss one.
- Bring your laptop to the appointment with your program up and ready to run in the terminal.

Throughout this course, you will be meeting with your TA to grade your assignment. To set up an appointment, follow the scheduling link on moodle.cs.colorado.edu for your section.



## Assignment 1 Grading Scheduler

The table below shows all available slots for an appointment. Make your choice by selecting a radiobutton and don't forget to click on "Save my choice" afterwards. If you need to make a change later you can revisit this page.

### Slots

| Date | Start | End | Location | Choice | TA | Group session |
|---|---|---|---|---|---|---|
| Thursday, September 4, 2014 | 9:00 AM | 9:10 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 9:10 AM | 9:20 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 9:20 AM | 9:30 AM | ECCS 102 | ◉ | Melissa Bica | Limited (1/1 left) |
| | 9:30 AM | 9:40 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 9:40 AM | 9:50 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 9:50 AM | 10:00 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 10:00 AM | 10:10 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 10:10 AM | 10:20 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 10:20 AM | 10:30 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 10:30 AM | 10:40 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |
| | 10:40 AM | 10:50 AM | ECCS 102 | | Melissa Bica | Limited (1/1 left) |

Meeting date and time slot

Room number

TA

Your sessions will last ~10 minutes, in which time you will run your program on your computer and show the TA the output. Please attend the grading meeting on time and there are no extra slots if you miss one. You will be asked a set of questions to test your understanding. These questions are not intended to trick you, only to gauge your understanding of the code and should be easy if you understand your code. If your code doesn't work, this is also a chance to discuss it with the TA. Your grade will be based on what you've submitted as well as your understanding of the concepts. A rubric for how your grade will be determined is available on moodle. This is what a typical grading rubric will look like:

**CU CS 1300 Assignment Grading Rubric**

**Name:** _____

**Date:** _____ **Assignment #:** _____

(This rubric is meant only as a guide. Actual grades may fall in-between these ranges.)

| Task Success (40%) | Solution Explanation (60%) |
|---|---|
| **40**<br>• Program performs all assigned tasks without error.<br>• Code compiles with no errors or warnings. | **60**<br>• Student clearly explains functionality of code during evaluation, and can answer all instructor questions.<br>• Explanations are clear and design decisions are well-justified. |
| **27**<br>• Program performs assigned tasks, but does not adequately handle all input.<br>• Code contains unresolved warnings or a small error upon compilation.<br>• Code fails to perform one or more assigned tasks. | **40**<br>• Student is able to explain their overall solution, but has difficulty answering specific questions.<br>• Student is not able to confidently justify programming decisions. |
| **13**<br>• Program has multiple compiler warnings or errors.<br>• Program has significant runtime errors or mishandles important aspects of the assignment. | **20**<br>• Student has trouble making comments about their own algorithms. Cannot answer a majority of the questions.<br>• Student offers little justification for decisions. |
| **0**<br>• Program does not run, or does not adequately perform any of its objectives.<br>• Unrecoverable errors exist in source code. | **0**<br>• Student cannot (or will not) explain decisions or answer any of the instructor's questions. |
| TOTAL: | |

**Final Grade:** _____ **/ 100**

This next section will be **DUE AT THE END OF RECITATION**.

Note: if you don't have a laptop, work with another person. When you are done with the exercise, show your TA that you both worked on the assignment together and you will both get credit for the work.

Name your program main.cpp.

Problem:

For this recitation exercise, you are going to write a program to prompt the user for the height of a mountain in feet (ft). Then, pass that height as a parameter to a function (named printFourteenerInfo) that will use the height to print out the name of the mountain and information about it.

| Height (in ft) | Mountain name |
|---|---|
| 14060 | Mount Bierstadt |
| 14036 | Mount Sherman |
| 14269 | Mount Antero |

The function should check the input height with the information provided in the table above and print information about the corresponding fourteener as shown below. If the height entered does not match any of the values, return the following message:

Height entered does not match a fourteener in this computer program. Please enter a correct height.

Here are the outputs for each of the mountains:

14060 ft is the height of Mount Bierstadt. It is near Idaho Springs.

14036 ft is the height of Mount Sherman. You can drive a long way to the trailhead if you're feeling lazy.

14269 ft is the height of Mount Antero. Apparently the hike is safe but it is one of the easier (i.e., boring) fourteeners.

Your program should use the following function template:

```
void printFourteenerInfo(int height) {
        //your code goes here
```

```
}
```

Your main() function should look like this:

```
int main() {
        //write code to ask for fourteener height

        printFourteenerInfo(height)

        return 0;
}
```

**To get credit for this lab exercise:**

- Please submit your main.cpp file to Moodle under the ‘Recitation 3 Submit’ link.
- Show the TA your code and run your program for them.