Data Structures & Algorithms in C

Prof. Georg Feil

Course Introduction

Summer 2018

Acknowledgement

- These lecture slides are partly based on slides and other material by Professor Magdin Stoica
- Additional sources are cited separately

Your Instructor (first half of course)

- Prof. Georg Feil
- email: georg.feil@sheridancollege.ca
  (Important emails from me go to your main Sheridan email, not SLATE!)
- My office: Davis B204
- My office hour: Thursday 3:00 pm – 4:00 pm
  (or other times by appointment, email me if you're coming!)
- My background:
  - Aerospace industry, robotics, astronomy
  - Real-time & embedded software, Unix/Linux (C and C++)
  - Android development (Java)

Our Goals for Today

- Get to know each other
- Understand the course
  - Structure
  - Requirements
  - What's expected of *you*
  - What's you can expect of *me*
- Start setting up the software you'll need on your laptop to write programs in C

Our Goals for the Entire Course (Learning Outcomes)

- This is an advanced course in algorithms and data structures
  - I assume that you have no previous experience with C
  - I assume that you are a good Java programmer including testing and debugging (e.g. Sheridan Java 1, Java 2 and Java 3 courses)
  - We'll be learning C, but not C++ (I may talk about C++ a bit)
- In this course, you will learn how to:
  - Develop efficient algorithms in C using proper programming techniques and standards
  - Identify and implement data structures required to solve specific problems
  - Dynamically manage memory to create and destroy data structures
  - Analyze complexity of algorithms including searching and sorting

Complexity Example

```
for (int i = 0; i < 1000; i++) {
    for (int j = 0; j < 1000; j++) {
        for (int k = 0; k < 1000; k++) {
            // Do something (takes 1 ms)
        }
    }
}
```

- How long will this take to run?
- What if you change all the '1000' to '10000'?

This course will challenge you!

- We'll have to learn C programming quite quickly
  - Not to worry... most of C is just like Java!
- We'll learn some advanced Computer Science concepts
- This course will be easy if you...

  Practice!

PROG20799 Course Evaluation

Midterm Test	30%	(week 7)
Final Exam	30%	(week 14)
Quizzes (5)	5%	(worth 1% each)
Assignments (4)	20%	(worth 5% each)
Project	15%	

- To pass the course students must:
  - *Average 50% or more on the midterm plus final exam*
  - *Average 50% or more overall*
- There may be additional in-class exercises or quizzes which don't count toward your final grade

Course Info

- For the official course outline see the link in SLATE, under Content > General
- My class plan with a week-by-week breakdown of activities is also available in SLATE
- If you need extra accommodation, e.g. extra time for tests, please come talk to me to introduce yourself
  - Register at Accessible Learning Services

Important Course Materials

- The recommended course textbook is "Advanced Topics in C: Core Concepts in Data Structures", by Noel Kalicharan, print ISBN-13 9781430264002
  - Get this version, not the one listed in the official course outline
  - Available free online

  http://proquestcombo.safaribooksonline.com.library.sheridanc.on.ca/book/programming/c/9781430264002

- The Kalicharan book doesn't have introductory material on C programming... for that we'll use this supplementary text: "C for Programmers", by Paul Deitel & Harvey Deitel, Prentice Hall / Pearson, print ISBN-13: 9780133462067

  http://proquestcombo.safaribooksonline.com.library.sheridanc.on.ca/book/programming/c/9780133462081

Additional Course Materials

- My course slides & notes, available on SLATE usually just before each class
- Notes you take in class
- **You will need to use all of these to succeed!**
  - Not everything will be in my slides

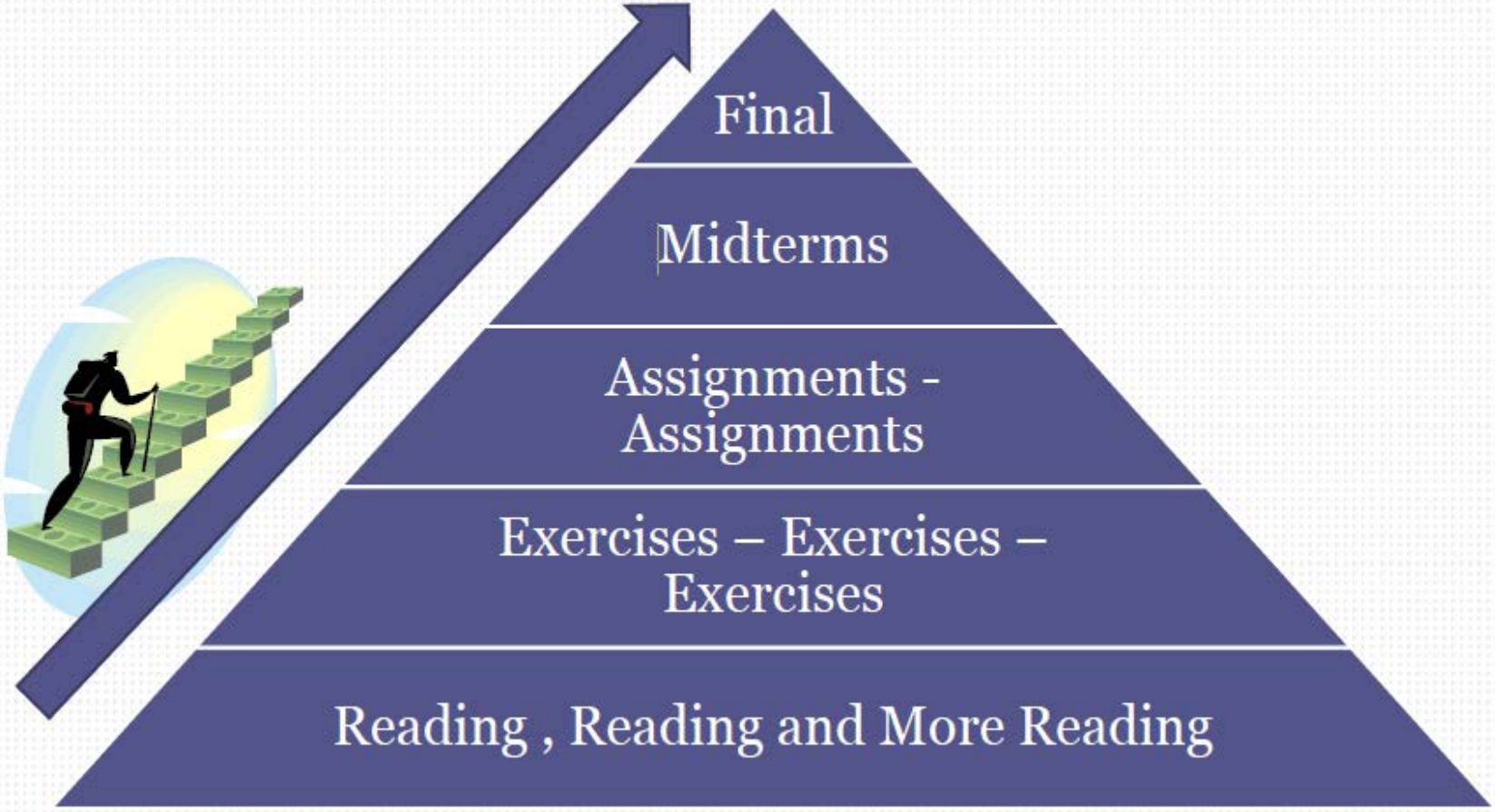
Data Structures Course Project

- This course includes a programming project
  - Think of it as a major assignment
  - Worth 15% of your final mark
- You'll get started on the project in week 8 (just after break week)
- You'll hand in the project in week 12 or 13
- While working on the project you'll also have to complete and hand in regular assignments
- More information later…

How to do well in this course…

- Programming is not something you learn just from a book or from lectures
- You learn programming by *DOING*!
- Work on assignments and exercises at home, spend time on them.
- Read the recommended textbook (assigned sections) and practice programming
  - You should spend at least 6 hours per week outside class on reading, exercises, assignments etc!
- We'll work on quizzes (1 – 5) and exercises in class so come prepared!
  - Laptop with required software installed

How to study in this course…



The cliffhanger / cramming approach doesn't work...

What Sheridan Expects of You

- All students are expected to follow the Sheridan Student Code of Conduct: https://www.sheridancollege.ca/-/media/files/www/life-at-sheridan/student-services/student-rights/student-rights-and-responsibilities/student-code-of-conduct-policy--01152015pdf-revised.ashx
- Strictly avoid plagiarism, copying, cheating
  - Everyone must view the library academic integrity tutorial: http://sheridancollege.libguides.com/academic_integrity
  - Watch the intro video, then click the "Next" button at the bottom for the rest of the tutorial
  - Shows useful examples of what is considered plagiarism or cheating
  - I'll be posting an additional academic integrity tutorial in SLATE

Academic Dishonesty (cheating/plagiarism)

- Sheridan has a formal Academic Integrity process and I will use it.
  - First offence: Mark of zero and a letter in your file
  - Second offence: Termination from the course (F/TM)
  - Third offence: Severe penalty, expulsion
- Major types of plagiarism
  - Copying or emailing assignments: DON'T DO THIS
    - Discussing ideas, concepts, and methods is OK
  - Copying from the internet: REWRITE text (sentences) in your own words, write code yourself
  - Your friend/uncle/mother writes assignment programs for you – DON'T DO THIS, you don't learn anything and may fail!

Who Copied From Who?

- It doesn't matter if plagiarism was on purpose or by accident
- It doesn't matter who did the work and who didn't
  - When plagiarism happens, all students involved are equally guilty!
- Never email or otherwise copy your work, or in any way allow it to be copied
- Don't be the "helpful" student who gives your work to others to "help" them
  - You will get a zero, maybe an F in the course, maybe get expelled
- Don't share passwords or mirror your hard drive using the cloud!

Online Resources

- Don't copy sentences/paragraphs from online sources, for example Wikipedia
  - Don't copy answers directly from any references, not even your textbook
  - Rewrite in your own words (and use proper English sentences)
  - Use more that one source of information
- Don't copy code from online sources, even open source sites like sourceforge
  - This is a programming course, you must write the programs yourself
  - *Copying example code from our textbook or my slides is OK*

Working Together and Helping Each Other

- I encourage students to help each other with assignments. When one student helps another in a productive way, they both end up understanding better.
- Some ways of helping are more productive than others, and some ways of "helping" are actually academic dishonesty – cheating.
- Good ways to help…
  - Talking things over with someone to help them understand a concept
  - Helping someone find the information they need
  - Testing another student's program to look for mistakes
  - Sitting with someone to advise them while they debug a program they are having trouble with
- Bad ways to help… (cheating)
  - Writing a part of somebody's code for them
  - Showing someone your code so they can write it down
  - Mailing somebody your program so they can use it as a template, cut and paste parts of it, or change it slightly and hand it in as their own

What I Expect of you in Class

- While in Class You Should NOT
  - Use your laptop to play games, watch YouTube, check email etc.
  - Work on tasks outside the scope of this class (for example assignments in other classes that are due)
- While I'm speaking or presenting slides
  - LISTEN, take notes
  - Don't use your laptop for anything not related to today's topic (close the lid)
  - Cell phones off, or in Airplane mode

What You Can Expect of Me

- I'm here to help you learn and succeed
  - Lend my experience and knowledge
  - Work together to overcome issues
  - This course should hopefully be interesting and fun!

- Feel free to email or arrange to meet me for any reason or concern you may have
  - Trouble with an exercise or assignment
  - Help with studying for quizzes & tests
  - Decisions related to the course
  - When in doubt, come to my office hour!

What if you have trouble on an assignment?

- *Don't* copy from your friends
- Come to my office hour (email first)
  - Time & location near the start of these slides
- Email me to arrange to see me another time
- Ask questions in class
  - Maybe 10 other people have the same question!

Let's Have Fun!

There are a lot of interesting things to learn…

Rest of this week:

- Learn about compiled computer programs and languages
- Start learning some C and data structure fundamentals
- Install C software development tools on your laptop
- "Hello World" program in C