

# Course Organization

---

**CPSC 501: Advanced Programming Techniques  
Fall 2020**

Jonathan Hudson, Ph.D  
Instructor  
Department of Computer Science  
University of Calgary

**Tuesday, September 8, 2020**



# Welcome!

## Jonathan Hudson, Ph.D

Lectures: Asynchronous (Zoom -> Content->Video, pre-recorded and available via Yuja)

Office: ICT 712 (I will not be in it!)

Office hours: 11:00-12:50 PM Mondays and Wednesdays (Zoom, link in D2L) or by email-scheduled appointments. (in between L01/L02, using waiting room, not recorded)

[jwhudson@ucalgary.ca](mailto:jwhudson@ucalgary.ca)

<https://pages.cpsc.ucalgary.ca/~hudsonj/CPSC501F20/>



# Tutorials

---

Start next week Monday.

Also through D2L, will be recorded, however point is active interaction with TA for material and assignment help.

Use link to your tutorial only.

Your enrollment tutorial TA will mark your assignment material and they are only responsible for the students enrolled in their tutorial.

# Why CPSC 501?

---

This course is full of things other courses likely never got to BUT are often encountered during more advanced programming and very useful to know.

# Course Goal

---

From the calendar:

- “Theory and application of advanced programming methods and tools. Recent issues as well as those of an enduring nature will be discussed.
- Topics may include the ***Fourier transform***; wavelets; functional programming; **genetic algorithms, simulated annealing and neural networks**; parallel and distributed programming; images and graphics user interface programming.”

# Course Outcomes

---

From the outline:

- Code refactoring, including how refactoring improves the design of software systems.
- Refactor object-oriented code using a disciplined process that uses version control and unit testing.
- Concepts of reflection as embodied in modern programming languages.
- Object introspection using the reflective capabilities of the Java programming language.
- Reflection serialization and deserialization using the reflection API of the Java programming language.
- Techniques used to optimize software performance, including how to measure performance gains.
- Optimize a computationally expensive digital signal processing program and create a report that describes the process they went through to accomplish this.

# Lectures

---

We will be using Java/Python in this course

We will cover:

- Refactoring
- Reflection
- Serialization
- Optimization
- Selected topics (AI methods, Digital Signal Processing)

# Out of lecture?

---

There is no attendance at tutorials but they are highly recommended

- Start Monday/Tuesday next week
- TAs will use classes to cover coding material in hands-on environment
- Material will be covered and there will also be assignment work/help

There is an D2L managed midterm scheduled

- Friday, November 6th, 2020 (24 hour period)



# Grading

---

Component	Weighting %
Assignments (4)	50%
Midterm	25%
Final	25%

- Each of the above components will be given a letter grade using the official University grading system. The final grade will be calculated using the grade point equivalents weighted by the percentages given above and then converted to a final letter grade using the official University grade point equivalents. (A+ are 4.3 for in-class component weighting)

# Example

---

- A1: A+
- A2: B
- A3: C
- A4: D
- Midterm: B+
- Final: A-

# Example

---

- A1: 4.3
- A2: 3
- A3: 2
- A4: 1
- Midterm: 3.3
- Final: 3.7

# Example

---

- A1: 0.5375
- A2: 0.375
- A3: 0.25
- A4: 0.125
- Midterm: 0.825
- Final: 0.925

# Example

---

- A1: 0.5375
- A2: 0.375
- A3: 0.25
- A4: 0.125
- Midterm: 0.825
- Final: 0.925
  
- Sum: 3.0375 (which is a B)

# Assignments

---

- Four individual assignments (50%)
- Refactoring, Reflection, Serialization, and TensorFlow Machine Learning
- Assignment deadlines will be 11:59 pm on Friday due dates.

# Exams (each 25%)

---

- Midterm
  - D2L – asynchronous (24-hour period)
  - Written (long answer)
  - Refactoring/reflection
- Final
  - Registrar scheduled via D2L (synchronous)
  - Written (long answer)
  - Cumulative
  - Light on selected topics
  - Focus on optimization and then midterm topics second

# Course Policies

---

- When you email include your first name, and last name.
- Please use “CPSC501F20” as the prefix in the subject line
- Make-up examinations and deferred examinations will not be provided except in cases of extreme personal emergencies.
- There are no late submissions. Submit early and double check after submitting. You can submit multiple times on D2L with no issue, so excuses will not be accepted.



# Academic Dishonesty

---

- *“A single offence of cheating, plagiarism, or other academic misconduct, on term work, tests, or final examinations, etc., may lead to disciplinary probation or a student's suspension or expulsion from the faculty by the Dean, if it is determined that the offence warrants such action.”*
- We have tools that let me quickly see if assignments appear to be highly similar and techniques like changing names, comments, and other details will not trick them.
- Please refer to the University Calendar for more details.
- **This course is fundamental and is essential for CS studies.**

# Academic Dishonesty

---

- *All the work you submit must be your own.*
- *When you take algorithms or segments of code from somewhere else you must cite where you obtained them from.*
- *You need to understand all of the code in your work because the midterm and final are evaluating your understanding, not if you were able to make it work*

# Be Computer Science 'Lazy'

---

- Search internet for answers.
- If you find something, read and understand it.
- Then develop your own solution using what you've learned.
- Do not copy and paste the answer! It is considered plagiarism!
- Being computer science lazy is an important skill and we will return to it.

# Getting Help

---

- Do your part: Attend the lectures and tutorials
- Act early!
- First try it yourself →
  - Study the material carefully
  - Break the problem down
  - Try to narrow down the question
  - Search on google for your answer
- Still unclear?
- Ask your TA
- Come to my office 😊

# Crisis line!

---

- If you think:
  - You suck at programming!
  - You suck at python!
  - You are not sure about this course!
  - You are OK with only a passing mark!!!
  - You tried but you didn't understand!
- Come to my office → I'll prove to you that you are wrong!
- Come early before things piled up!

# Textbooks

---

- Recommended Textbook(s):
  - Martin Fowler, Refactoring: Addison-Wesley.
  - Forman and Forman, Java Reflection in Action: Manning Publications.

# Access to CPSC

---

- The Java 8/Python 3 are your primary work environment for this course.
- Assignments & exercises are acceptable if they run on Java 8 (1.8) and Python 3.6.8+
- You can access the CPSC lab remotely. (no need to do this)
  - SSH (Secure Shell) allows you to establish a remote connection with the CPSC lab.
  - [https://ucalgary.service-now.com/it?id=kb\\_article&sys\\_id=29aedd1bdb3e63c0d1b63ccb7c961963](https://ucalgary.service-now.com/it?id=kb_article&sys_id=29aedd1bdb3e63c0d1b63ccb7c961963)
- Please do not use any non-Linux-based CPSC server for this course.

# Onward to ... Refactoring.

---

Jonathan Hudson  
[jwhudson@ucalgary.ca](mailto:jwhudson@ucalgary.ca)  
<https://pages.cpsc.ucalgary.ca/~hudsonj/>



UNIVERSITY OF  
CALGARY