Introduction To CPSC 231

James Tam

Administrative (James Tam)

- Contact Information
  - Office: ICT 707
  - Email: tamj@cpsc.ucalgary.ca

- Office hours
  - Office hours: Monday (11 - 11:50 AM after 233 lecture), Wednesday (3:00 - 3:50 PM after my 231 lecture).
  - If I'm not in my office give me a few minutes or check the lecture room.
  - Email: (any time)
  - Appointment: email, phone or call
  - Drop by for urgent requests (but no guarantee that I will be in if it's outside of my office hours!)

My Office

Course Resources

- Required resources:
  - Course website: http://pages.cpsc.ucalgary.ca/~tamj/231 (Get the notes off the course webpage before lecture)

- Recommended but not required:
  - "*The Practice of Computing Using Python*" by William Punch, Richard Enbody (available in the bookstore).
  - "*Learning with Python*" by Jeffrey Elkner, Allen B. Downey and Chris Meyers (free book available online or pre-printed copies available for purchase).

How To Use The Course Resources

- They are provided to support and supplement this class.

- Neither the course notes nor the text book are meant as a substitute for regular class attendance.
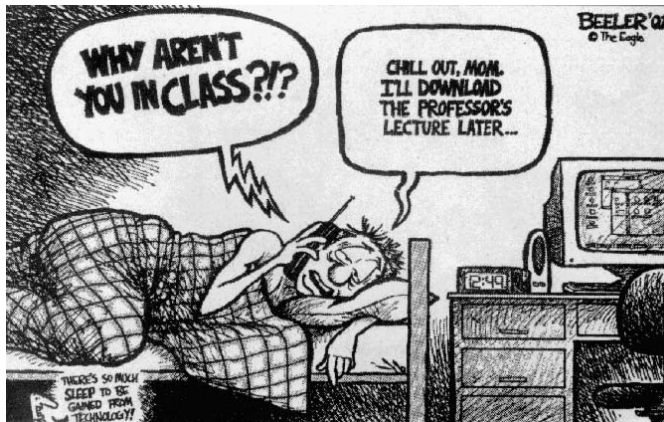
How To Use The Course Resources (2)

```python
def display (world):

    sys.stdout.write(' ')
    for i in range (0, columns, 1):
        if (i < 10):
            print i,
        else:
            num = i + 55
            ch = chr(num)
            print ch,
    print

    for i in range (0, columns, 1):
        sys.stdout.write(' -')
    print

    for r in range (0, rows, 1):
        for c in range (0, columns, 1):
            sys.stdout.write('|')
            sys.stdout.write(world[r][c].appearance)
        print ('|'), r
        for i in range (0, columns, 1):
            sys.stdout.write(' -')
        print
```

If you miss a class make sure that you catch up on what you missed (get someone's class notes)

...when you do make it to class make sure that you supplement the slides with your own notes (because you aren't going to remember it in the exams if you don't)

How To Use The Course Resources (3)

• What you are responsible for:
  - Keeping up with the content in class which includes the topics covered but also announcements or assignments whether you were present in the class or not.
  - If you are absent, then you are responsible to get the information from the other students in class.

• However, after you've caught up by talking with a classmate:
  - Ask for help if you need it
  - There are no dumb questions

A Bit About CPSC 231

• It is a course geared primarily towards CPSC majors.

• But it is not assumed that you have prior knowledge of Computer Science.

• It can be a lot of work.

How To Succeed

•Successful people

Leonardo Da Vinci

Bruce Lee

J.R.R. Tolkien

Amadeus Mozart

Wayne Gretzky

How To Succeed In This Course

1. Practice things yourself.
   - You get better by doing things for yourself (this is a 'hands-on' field of study and work).
   - Write lots programs.
     • At the *very least* attempt every assignment.
     • Try to do some additional practice work (some examples will be given in class, some practice assignments will be available on the course web page).
     • Write lots of little 'test' programs to help you understand and apply the concepts being taught.
   - Trace lots of code (computer programs)
     • Reading through programs that other people have written and understanding how and why it works the way that it does.

How To Succeed In This Course (2)

2. Make sure that you keep up with the material
    - Many of the concepts taught later depend upon your knowledge of earlier concepts.
    - Don't let yourself fall behind!
    - *At least* attempt all assignments!

```
            :
     Problem decomposition
      Loops/repetition
     Decisions/branching
  Introduction to programming
```
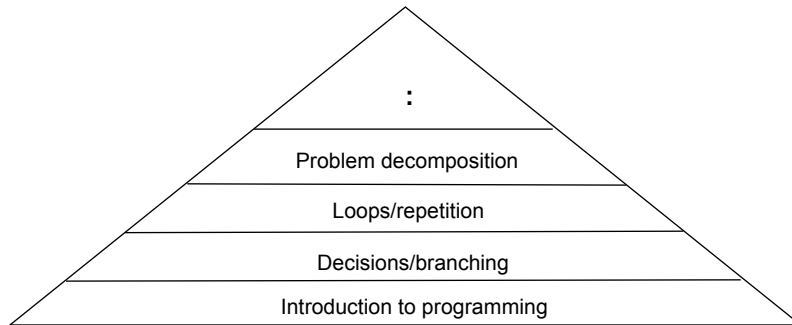
How To Succeed In This Course (3)

3. Look at the material before coming to lecture so you have a rough idea of what I will be talking about that day:
    a) Read the slides
    b) Look through the textbooks

How To Succeed In This Course (4)

4. Start working on things as early as possible:
   - Don't cram the material just before the exam, instead you should be studying the concepts as you learn them throughout the term.
   - It's important to work through and understand concepts *before* you start assignments. If you try to learn a new concept and work out a solution for the assignment at the same time then you may become overwhelmed.
   - Don't start assignments the night (or day!) that they are due, they may take more time than you first thought so start as soon as possible.

How To Succeed In This Course: A Summary

1. Practice things yourself
2. Make sure that you keep up with the material
3. Look at the material before coming to lecture
4. Start working on things early

Computer Science Is About Problem Solving

- A simplified description of what this means: Write a computer program that performs a task (fulfilling a need and thus solving a problem).

- This requires that you know how to write a program in a given language but goes beyond knowing the rules and structure of a language (this is the problem solving aspect).

- For example you may know how to get a program to rerun itself (loop) but you may not know how to loops are applied to a given a problem.

- You get better at problem solving through practice ("How to succeed in this course").
  - This is why lectures won't directly address the solution to an assignment.

Computer Science Is About Problem Solving (2)

- There isn't an exact prescribed formula or series of steps that you can learn and apply.

- But you aren't left alone to fend for yourself!

Computer Science Is About Problem Solving (3)

• You will be taught:
- Computer programming: The mechanics of how different programming concepts work e.g., file input and output, displaying text and graphics onscreen.
- Problem solving strategies: approaches to creating a problem to a challenging solution:
  • Practice! Practice! Practice!
  • Example strategy: Problem decomposition.
  • Example strategy: Visualization techniques.
  • Good programming style.

Course Goals

• Understand basic programming constructs such as branching and looping.

• Develop basic problem solving and analysis skills.

• Being able to implement a solution in a procedural programming language for a moderately sized problem using good design principles.

Feedback

What is he talking about???

Wow I am the greatest speaker in the world!

Let me know how things are going in the course:

- Am I covering the material too slowly or too quickly.
- Can you read the slides and my hand writing.
- Can you hear me in the class.
- Etc.

James Tam