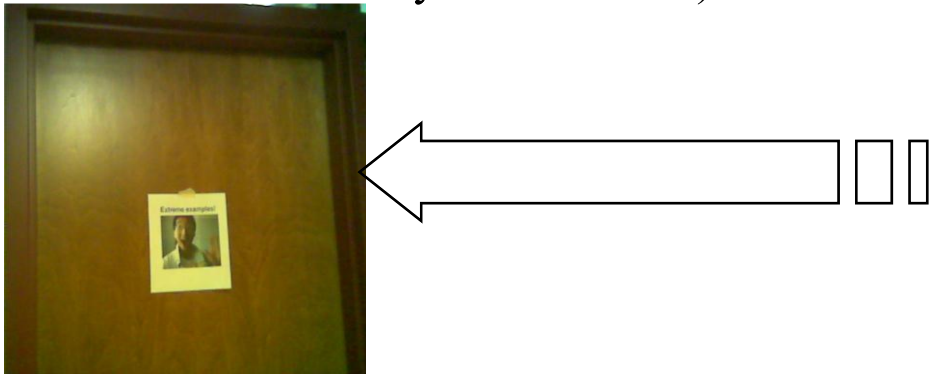
Introduction To CPSC 231

James Tam

Administrative (James Tam)

- Contact Information
  - Office: ICT 707
  - Email: tamj@cpsc.ucalgary.ca
- Office hours
  - Office hours: Monday (3:00 – 3:50 PM), Tuesday (2:00 - 2:50 PM)
  - 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!)



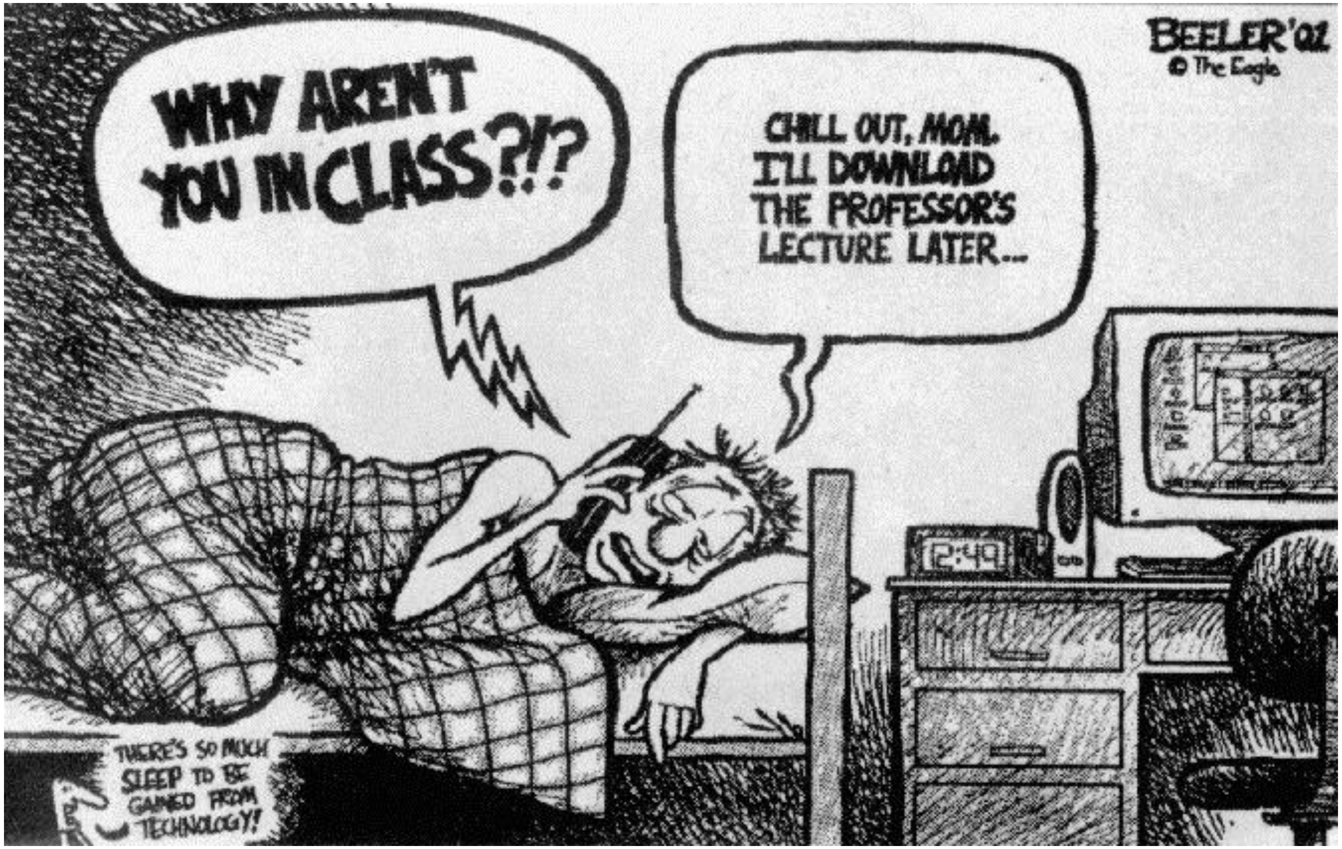
My Office

Course Resources

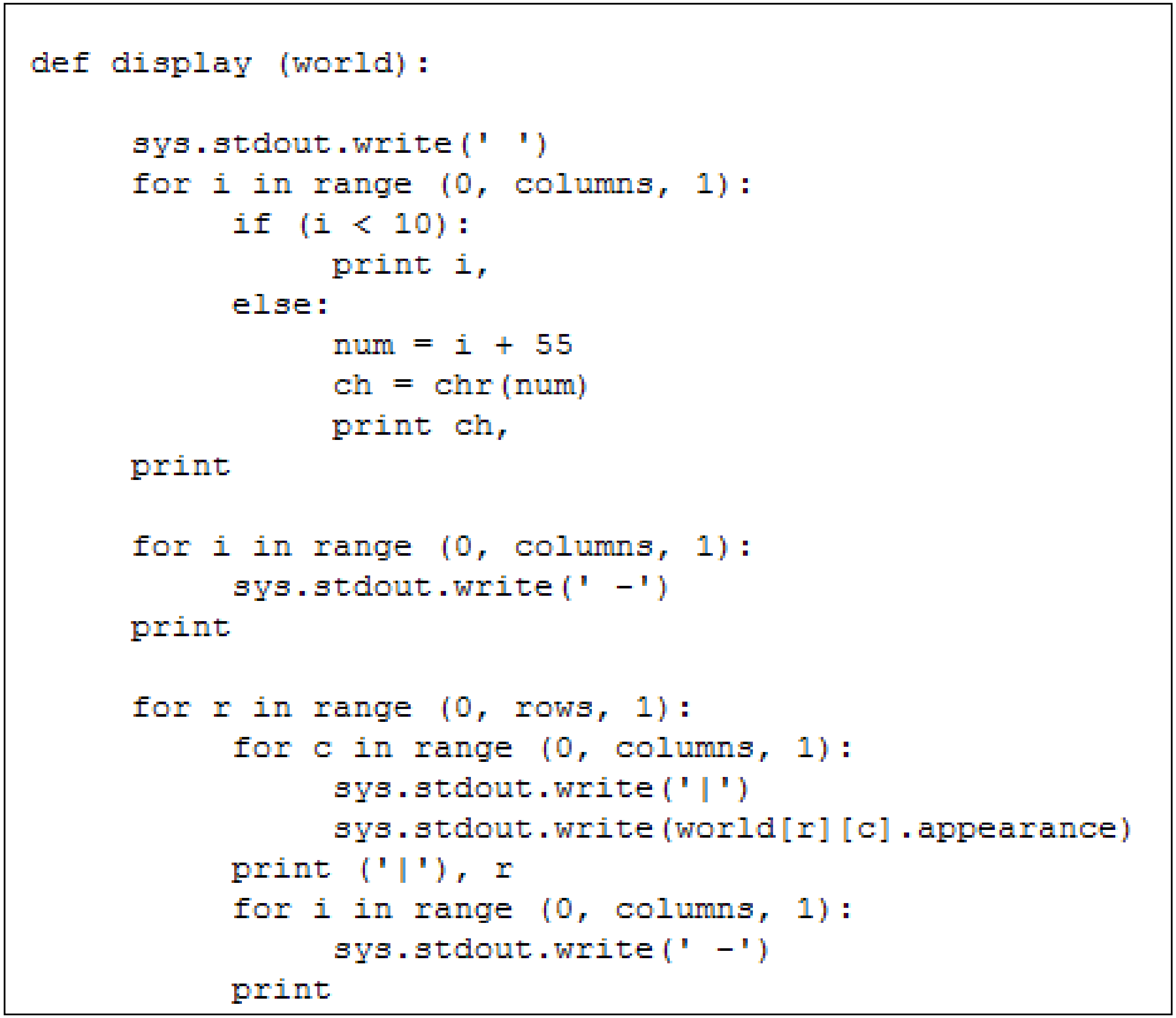
- Required resources:
  - Course website: http://pages.cpsc.ucalgary.ca/~tamj/231 (Get the notes off the course webpage before lecture)
- Recommended but not required:
  - Python: Visual QuickStart guide (2nd Ed): Licensed for no-charge online access to students and staff via the library website: http://proquest.safaribooksonline.com.ezproxy.lib.ucalgary.ca/

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.


BEELER'02
© The Eagle
WHY AREN'T YOU IN CLASS?!?
CHILL OUT, MOM. I'LL DOWNLOAD THE PROFESSOR'S LECTURE LATER...
12:49
THERE'S SO MUCH SLEEP TO BE GAINED FROM TECHNOLOGY!

How To Use The Course Resources (2)



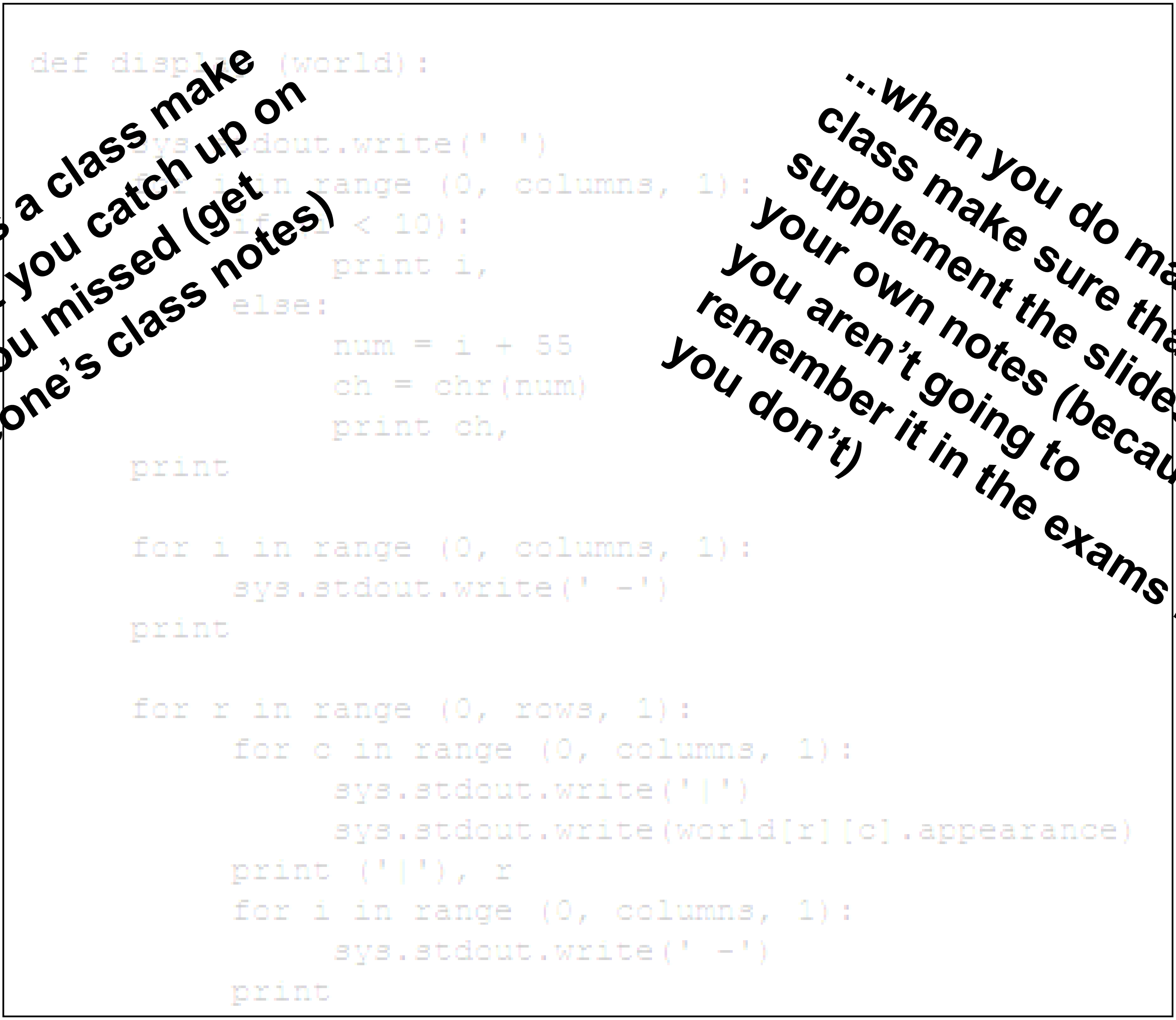
```
def display (world):

    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
```

How To Use The Course Resources (2)



```
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 for getting the information from the other students in class.
  - (I won’t be able to repeat the lecture content if you are absent…there’s just too many of you to make it practical).
- 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.

Wav file from “The Simpsons”

Common Interview Questions

- Besides looking at degrees granted and grades received, many tech companies (e.g., Google) may ask you questions that appear non-technical:
  - You’re asked to solve puzzles during the interview.
- There is a relationship between skill at solving puzzles (“problem solving”) and success in a (technically oriented) industry.

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…how do you *apply your knowledge and skills*).
  - Analogy: you may the rules and structure required to produce a poem but it takes more than that to write good poetry.
    - "Ode to my keyboard"….
- For example you may know how to get a program to run across the Internet but you may not know how to write a good game app on Facebook™.
  - "*This *%$#! App really sucks!*"

Computer Science Is About Problem Solving (2)

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

Computer Science Is About Problem Solving (3)

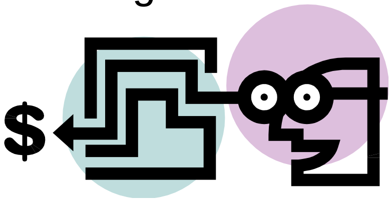
- There isn’t an exact prescribed formula or series of steps that you can learn and apply.

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

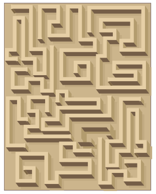
Computer Science Is About Problem Solving (4)

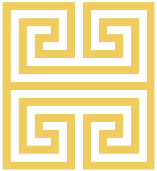
- You will be taught:
  - Computer programming: The mechanics of how different programming concepts work e.g., How do you get a program to read from a file.
  - Problem solving strategies: approaches to creating a solution to a challenging problem:
    - Practice! Practice! Practice!
    - Example strategy: Problem decomposition.
    - Example strategy: Visualization techniques.
    - Good programming style.

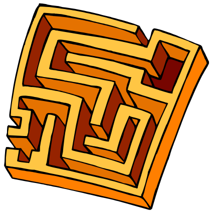
Providing solutions to assignments may be popular to students but useless for learning



What's needed is for me to teach you the skills to solve any reasonable problem







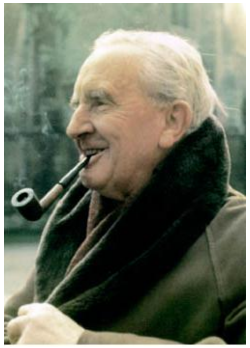
Course Goals

- Understand basic programming constructs such as branching and looping.
- Develop basic problem solving and analysis skills.
- Being able to implement a solution for a moderately sized problem using good design principles.

How To Succeed

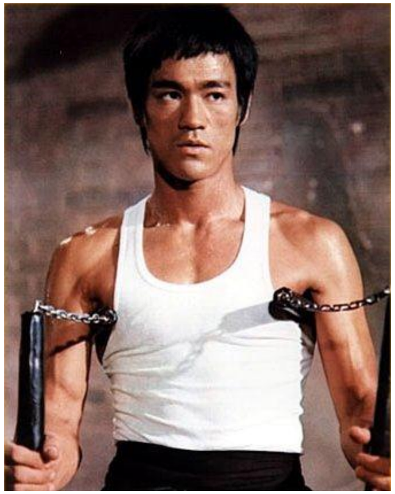
- Successful people

Leonardo Da Vinci



J.R.R. Tolkien

Amadeus Mozart



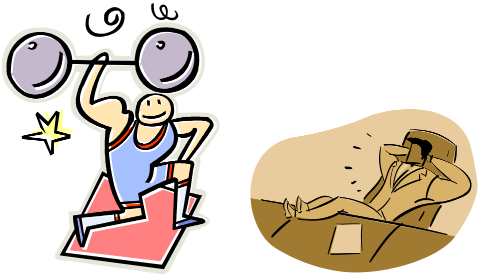
Bruce Lee

How To Succeed In This Course

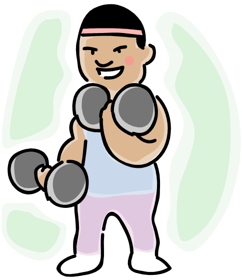
1. Practice things yourself.
   - "I wish he [JT] would help us more by giving us code [parts of a computer program] that can be directly used in the assignment."

   - How Computer Science works: You get better by doing things for yourself (this is a 'hands-on' field of study and work).

Similar to getting fit: you can't just watch



You have to do it yourself

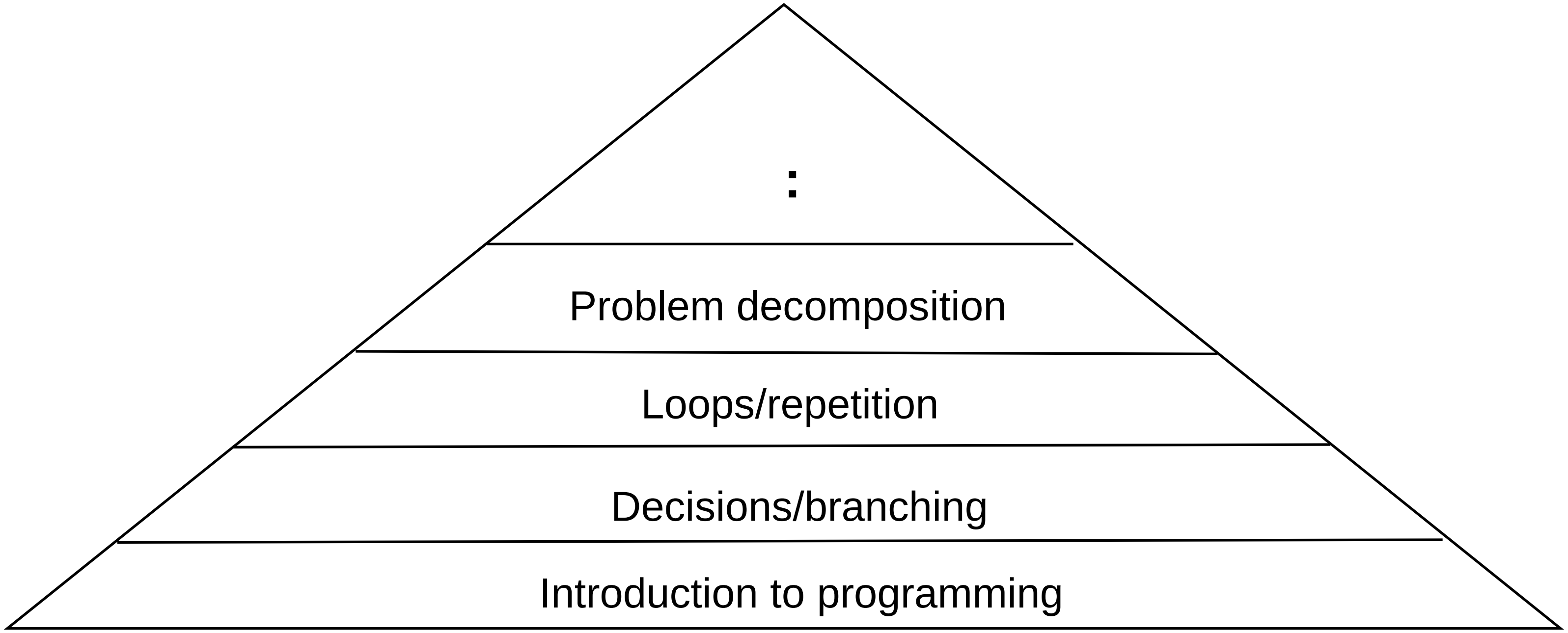


How To Succeed In This Course (2)

- 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 executing it ‘by hand’ in order to understand how and why it works the way that it does.

How To Succeed In This Course (3)

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!



How To Succeed In This Course (4)

- If you find concepts unclear trying to understand them on your own can be beneficial (because this is a ‘hands on’ field).
  - Read alternate explainations of the concepts covered in class in the text book (or other textbooks: remember that electronics books accessible through the library (Safari) are ‘free’).
  - Looking at online resources:
    - Remember academic resources just like other online information may not be a good source.
    - Start with more reputable sources e.g., www.python.org
- Addendum to the previous point #2 and a point raised earlier “ask questions”.
  - If you are still unclear on concepts then make sure that you ask for help.
  - Don’t wait too long to do this because latter concepts may strongly depend on the understanding of earlier concepts.
  - (If your first time that you come for help is the last week of the term or worse after the end of term then it’s probably far too late).

How To Succeed In This Course (5)

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 textbook(s)

How To Succeed In This Course (6)

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

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.