Introduction To CPSC 231

- Procedural programming
- The python language
- Problem solving

James Tam

Contact Information (James Tam)

- How to reach me
  - Email: tam@ucalgary.ca
  - Make sure you **specify the course name and number** in the subject line of the email 'CPSC 231' (otherwise I might miss it).
- Office time
  - Tuesday, Thursday: 4:00 – 4:30 PM
  - Zoom (passcode = hope) URL: https://ucalgary.zoom.us/j/98406536985
  - In D2L: Content->Course information->Help resources and contact information
  - Other help resources are available:
    - In D2L look under: Content->Course Information
    - Scroll down until you see the heading: "Help resources and contact information"

James Tam
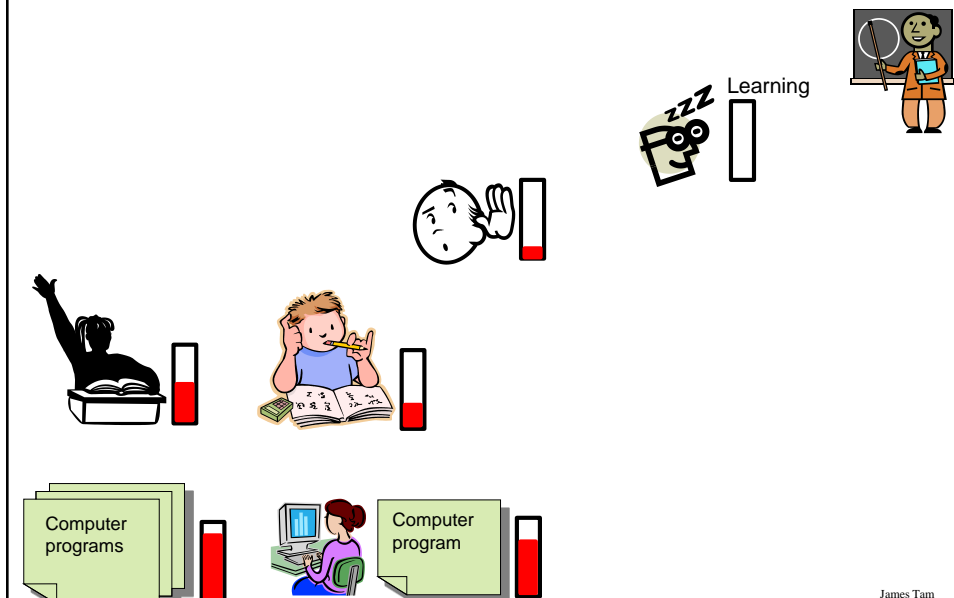
Course Resources

- • Required resources:
  - D2L site (You **must** get the notes from here before lecture)
    - ○ Course website can be reached via D2L (`Content->Lecture`) or
      https://pages.cpsc.ucalgary.ca/~tamj/2025/231P/
      - ▪ The website includes all resources (not all of which may be individually linked to in D2L(and may display better than viewing them from within D2L.
      - ▪ Recommended but not required textbook:
        "Python for Everyone" (Wiley) 3rd edition (eText or paper)
  - Alternatively you can access any book licensed by the university ("for free") on the library web site:
    - ○ https://library.ucalgary.ca

Your Engagement Level -±-> Your Learning

Learning

Computer programs

Computer program

Lecture: How To Use The Course Resources

• They are provided to support and supplement this class.

- The notes outline the topics to be covered

- *At a minimum* look through the notes to see the important topics.

- However the notes are just an outline and just looking at them without coming to class isn't sufficient to do well

- You will get additional details (e.g., explanations) during lecture time

  • Take notes!

  • If you miss a lecture then get a copy of the in-class notes from another student (who takes detailed notes)

James Tam

Lecture Content

• You need to attend class

• New programming concepts
  - E.g. repetition, functional decomposition…

• Filtering concepts (i.e. you are told what to focus on): programming languages are extensive, even professional software developers aren't experts in all parts of a language.
  - You will be informed as to what concepts are important for this course
  - What parts of the language are important.
  - How to do things with language e.g. how to use a random number function as computer simulation.

• While "Googling it" can provide links to example programs Google won't show you all the answers e.g. what's important for this course.

• Beyond this lecture will provide some opportunities to develop your skills e.g. we'll go over practice problems

James Tam

Tutorials

- There's two types: teaching and help tutorials.

- **Both of them will be in person**.

- Teaching tutorials (you registered in a specific section when you signed up for the course).
  - Similar to lecture, again you should be attending class and catching up if you miss a class.
    - Reinforce concepts but the audience size is smaller (easier to ask questions about content) and hearing an alternative explanation for complex concepts can be beneficial.
    - The teaching assistants will also go over assignments.
    - Some additional hands on work (exercises) will allow you to develop your skills.

Tutorials (2)

- Help tutorials (CT/Continuous tutorial)
  - Attendance is optional.
  - It's your opportunity to ask questions.
  - Located in the computer lab (near MS151 – there's a long table with computers).
  - Staffed by the teaching assistants (who teach the tutorials):

T01: MW 12:00 - 1:50 PM MS 160 Danissa Sandykbayeva (Windows user) danissa.sandykbay1@ucalgary.ca

T03: TR 10:00 - 11:50 AM MS 160 Mika Ong (Windows user)   mikaelahope.ong@ucalgary.ca

Tutorial: Days/Times & Instructor Information

- In D2L:
  - Content->Teaching tutorials
- Direct web address:
  - https://pages.cpsc.ucalgary.ca/~tamj/2025/231P/#Tutorial_information_ (teaching_and_help_tutorials)

Tam's House Rules (Remote Learning)

- Some of the common sense rules and social norms that apply for face-to-face to learning also apply to the online version e.g. turn taking.
- Please do ask questions!
- But when you have a question use the "raise your hand" feature in Zoom: Reactions -> Raise hand
  - Of course you shouldn't just turn on your microphone and start talking.
  - Please raise your hand and wait to be acknowledged before you use your talk (with your mic) or type into the chat.

Tam's "House Rules" (In Person)

- I will endeavor to keep the lecture within the prescribed time boundaries

- You won't pack up and end before time is up

James Tam

Tam's "House Rules" (In Person)

- No recordings/captures without permission during class please

- (Recall that learning tends to increase with additional levels of engagement).

James Tam

Tam's "House Rules" (In Person)

• Quiet whispering is OK…

…but make sure if it is *quiet*. If it's loud enough for me to hear then it's likely that others are being disturbed by the noise as well.

Images from colourbox.com

James Tam

Yes: "This Stuff Will Be On The Exam"

• The administrative notes contains important information e.g. how your grades are calculated, course policies etc.

• To encourage students to pay attention to details (and to reward those who do so):
  - **Some of your midterm multiple questions will come from the section dealing with administrative information**.
  - You may see a question or two from this section on the final exam as well.

James Tam

Student Questions?

- Q: How do I know how I did on an assignment? Can we get the grading for previous assignments before the next assignment is due?
- A: The two questions are actually inter-related.
  - o **Test your program:**
    - With all assignments you will be graded according to the number of program functions that are correctly implemented ("program functionality").
    - The required program functions along with grading weights will be specified beforehand.
    - If you test your program thoroughly before submitting the final version then you should get a pretty clear idea of "how you did".
  - o **Familiarize yourself with style and documentation requirements:**
    - Again these requirements will be listed in the assignment description.
    - Details will be provided throughout the semester but these requirements are first introduced in the "Introduction to computer programming" components.
  - o **Even though the marking for an earlier assignment may not be available prior to the due date of the next assignment** you should have the ability to know "how you did".

James Tam

Implementing Assignments

- You will create a working and executable computer program.
- Use a text editor (similar to a word processor minus the fancy formatting capabilities) to create it and you will electronically submit the text file (to D2L) for marking.
- **The official editor that we will support for this class was developed by the same people who developed the python language: IDLE** (with the Windows installation for python automatically comes with download and install).
- Although you may be given some time in tutorial to work on your assignments (during the "open tutorial") mostly you will complete your work on your own time.
  - o Don't underestimate the time/effort required.
  - o Creating a good working program is harder than it may first appear.
- But on the flip side previous students have "gotten through the course" with no prior programming experience if they approached the course properly.

James Tam

The Programming Language For Assignments

- All evaluation components must be completed by writing a program using Python version 3.x (not version 2.x).

- **If your program doesn't work under this condition then it will not be marked.**

- Installing python on your own device isn't mandatory but here's some information:
  - https://pages.cpsc.ucalgary.ca/~tamj/2025/231P/notes/pdf/installing_accessing_python.pdf

Collaboration

- Each student must work on his/her own assignment (no group work is allowed for this class)

- Things you **should NOT** do for this class:
  - Do not communicate in any form your assignment solution:
    - Students must not see each other's solutions for assignments.
    - Students should not discuss assignment solutions.
    - (Also you should not be communicating your solutions to other graded components, such as examinations, with other students).
  - Do not go over the assignment with your tutor (if you employ one).
  - Do not use code from external sources e.g. books, web, A.I. tools etc.
    - You can use examples from lecture and tutorial however (that's why they were created).

- Violating these rules may result in an academic misconduct investigation being headed by the office of the dean.

- Additional details will be provided later during the semester.

10 Mini Assignments: 0.5% Each

- As the name implies mini-assignments are a smaller version of the (full) assignments.
- The purpose is to allow you to apply the technical concepts (e.g., branches, functions, loops etc.) without the real-world complexities of a full program.
- Marking will focus on 'functionality': getting the program features/functions to work.
  - However some mini-assignments may have specific design requirements so make sure you are familiar with each assignment's requirements.
- Although you shouldn't ignore other things such as style and documentation these things won't be graded for the mini-assignments

4 Full Assignments

- You will write a larger and more challenging than the mini-assignments.
- Marking will not only be based on the functionality of your program (i.e. does it work) but other criteria such as programming style and documentation (additional details will be provided during the semester as each assignment is released).

Component	Weight
A1	8%
A2	8%
A3	10%
A3	10%
A4	7%
Midterm exam	25%
Final exam	37%
10 Mini assignmentsx0.5% each	0.5%
Total	100%

Assignment Grading

- Assignments will be marked by the tutorial instructor who teaches your tutorial section.

- When you first see your grade in D2L: If have questions or concerns about grading **the tutorial instructor** (in the tutorial that you are registered) **is the person to contact**.
  - For your convenience the link to tutorial information has been provided again: https://pages.cpsc.ucalgary.ca/~tamj/2024/231P/#Tutorial_information_(teaching_and_help_tutorials)
  - Your tutorial section was the one you were required to select when you registered.
  - If you don't know how to find that information then you can find it the "Student Center" in PeopleSoft:
    - https://cas.ucalgary.ca/cas/login?service=https://portal.my.ucalgary.ca/psp/paprd/?cmd=start&ca.ucalgary.authent.ucid=true

James Tam

Submitting Assignments

- Submit your work to the appropriate D2L Dropbox link.
  - http://d2l.ucalgary.ca/
  - Find the appropriate course name/number for the semester.
  - Within the D2L Dropbox setup for the course there's a separate link for each assignment.

- **Bottom line: it is each student's responsibility to make sure that the correct version of the program was submitted on time.**
  - Alternate submission mechanisms e.g., email, uploads to cloud-based systems such as Google drive, time-stamps, TA memories cannot be used as alternatives if you have not properly submitted into D2L
  - **Only files submitted into D2L by the due date is what will be marked**

James Tam

JT's Hint: Electronically Submitting Work

- Bad things sometimes happen!
  - A virus, hardware failure, you screwed up the submission.

- Rules of thumb for assignment submissions:
  - Do it early! (Get familiar with the system).
    - You can submit multiple times in D2L.s
  - Do it often! (If somehow real disaster strikes and you lose everything at least you will have a partially completed version that your TA can mark).
  - *Check your work*.
    - Don't assume that everything was submitted OK.
    - Don't just check file names but at least take a look at the actual file contents (not only to check that the file wasn't corrupted but also that you submitted the correct version).

James Tam

How To Verify Submissions In DropBox

- There is a help link provided with each assignment description.

- When Teaching Assistants go over Assignment zero you will have the opportunity to work through the assignment.

- Assignment zero:
  - Not graded but important practice
  - Learning objectives:
    - How to submit assignments using D2L (most know this).
    - **How to properly check if a submission "went through"** (main learning objective as most students do not do this).
    - Learning about file name extensions: how to view and change them (needed to send 'restricted' file attachments such as python programs).

- Resource file for checking submissions:
  - http://pages.cpsc.ucalgary.ca/~tamj/resources/Verifying_D2L_Submissions.pdf

James Tam

Backing Up And Submitting Your Work

- Bottom line: **it is up to you** to make sure things are done correctly and on time.

- If you have questions beforehand then do ask (make sure you ask your questions early enough so you can receive an answer before the due time).

- But don't wait until after the due date (it's too late).
  - If your work isn't in D2L before the final due date then **you will be awarded no credit**.

Assignments: Late Submissions

- If you have a legitimate reason (and approved documentation) for an extension to get in touch with the course instructor (tam@ucalgary.ca) **before the deadline** (don't wait until after the due time/day).

- **Full assignments**: Late submissions without an instructor approved extension will have the following penalties applied.

Submission received:	On time	Hours late : >0 and <=24	Hours late: >24 and <=48	Hours late: >48 and <=72	Hours late: >72 and <=96	Hours late: >96
Penalty:	None	-1 GPA	-2 GPA	-3 GPA	-4 GPA	No credit

- **Mini-assignments**: can be at most submitted one day late:

Submission received:	On time	Hours late : >0 and <=24	Hours late: >24
Penalty:	None	-1 GPA	No credit

Examinations

- There will be two examinations: midterm and final exam.
- Midterm exam
  - **In person midterm** **Thursday May 29 2:45 – 3:45 PM (location TBA).**
- Final exam (again for multi-section: common to all lectures, out of class)
  - Date/time/location determined by the Office the Registrar.
  - You can find information about your final exams online via the university PeopleSoft portal.
- Both exams will be completed on paper (not in front of a computer).
- **Extra requirement on top of the raw score (comes from the official & approved outline from science):**
  - You must pass the weighted average of the exam component to be awarded a grade of C- or higher in the class.

James Tam

Examinations (2)

- Information about the examinations will be available on the main grid before the respective test:
  - In D2L: Content->Examination information
  - Under the main index:
    - o Main grid: Course topics, lecture notes, assignment descriptions, exam information

Index (shortcut to major sections of the webpage)

- A brief overview of the programming language you will learn
- Contact information:
  - o Course instructor
  - o Tutorial instructors
- Course administration: grading information, textbooks, instructor contract information
- Tutorial information: scheduled teaching tutorials along with tutorial examples, drop in help tutorials (in both cases the schedules can be found here)
  - o Getting help from tutorial instructors via the Help tutorials/Continuous Tutorials (CT)
- Main grid: course schedule for the lecture, lecture notes, assignment & exam information

James Tam

Examination Content

- Multiple choice questions:
  - Partial program traces e.g., what's the program output
  - Basic program structure e.g., find the errors, which function or operator is needed for a particular mathematical operation
  - More examples and details coming during the semester

- Written questions:
  - Write a small/partial computer program.
  - Trace the execution of a computer program e.g., what is the 'output'.
  - Conceptual (lower weight for this type of question) e.g., definition of a technical term.
  - Likely there will be a smaller proportion of written questions on the midterm vs. the final.

- I will be grading the exams.
  - (I'll do the best I can to get them done in a timely fashion but remember there's a fair number of you in the class).

Examination Content (2)

- More sample 'exam type' questions will be provided during the semester.
  - Sometimes 'on the fly' in lecture so **pay attention** to these and **take notes**.

Grades For Each Component

- The official grading mechanism for this (and most) universities is a letter grade/grade point e.g. A/4.0, A-/3.7 etc.

- Term grades must be stated as a letter grade.

- Component grades (assignment, exam etc.) can either be a letter grade or a raw score (e.g. percentage)

- For this class
  - **each major component will be awarded a grade point (and not a percentage) e.g. the 2.0 GPA and not 65% will be used to calculate your term grade.**
  - and this is the value used to determine the term grade.
  - If want to know the reason why grade points are used for this class: [Information link: https://pages.cpsc.ucalgary.ca/~tamj/2025/231P/notes/pdf/why_are_grade_points_used.pdf

---

Grading: Course Components

- Each course component will have a grading key

- Sample from a past assignment (different course)

> 5. Print the document (you won't actually be able to print anything in the 203 labs because there are no printers connected to the computers) but your program should be able to invoke the print command using VBA. (**+0.2 GPA**)
>
> 6. Close the document and automatically save changes (no choice given to the user). (**+0.2 GPA**)
>
> 7. Instead of applying Features 1 - 6 on just a single document, the macro will instead it will prompt the user for a location (e.g., "C:\temp") via a InputBox and apply Features 1 - 6 to every Word document in that location. When you write the program you can assume that the folder only contains Word documents. You must employ nesting in order to get credit for this feature, an outer loop successively opens each document in the specified location and inside the loop body Features 1 - 6 will be applied. (**+1.0 GPA**)

  - A student who completes only Features #5 – 7 will be awarded an assignment grade point of 0.2 + 0.2 + 1.0 = 1.4 (just over a D+ grade)

- Examinations will include a lookup table (raw percentage score to grade point). Sample from another course.

Min percentage	GPA	
85.00	3.8	
86.00	3.9	
87.00	4	A
94.00	4.1	
96.00	4.2	

Grade Points Are Letter Grades Not Percentages

- For examinations the mapping between a raw score and a grade point occurs one way (raw score mapped to grade point)
  - Example (purely for illustration purposes) 65 – 69% = C/2.0, 70 – 74% = C+/2.3
  - But **grade points don't map back to percentages**
    - E.g. I was awarded a 66% on midterm and then I see this is a 2.0 GPA (out of 4.0)
    - Does this mean that my percentage 'went' from a 66% to a 50%!!!???
    - No.
      - A 'C'/2.0 does not mean that 50% was awarded as a course grade.
      - To put this in perspective a passing grade point in this university is a 1.0/D in a course. If a grade point mapped back to a percentage this would mean that anyone getting a 25% or higher would pass any course here.
- The mapping of the midterm to grade point will be posted sometime after the midterm grades have been released.
- The mapping of the final exam to grade point will be posted sometime after the final exam grades have been released.

James Tam

Calculating Your Overall Term Grade Point

- To determine your weighted term grade point simply *multiply each grade point* by the weight of each component.
  - **Percentages won't be used to determine the term grade point/letter**
- Sum the weighted grade points to determine the term grade.
- Simple and short example (not exactly the same as this term but it should be enough to give you an idea of how to do the specific calculations required this semester):
  - Assignment 1: weight = 60/100, example score = A
  - Assignment 2: weight = 40/100, example score = B+
  - Weighted assignment 1: 0.6 * 4.0 = **2.4**
  - Weighted assignment 2: 0.4 * 3.3 = **1.32**
  - Total term grade point = **2.4 + 1.32 = 3.72**

  (In this case the term letter is A- if the official university cutoffs were used – more on this shortly)

  (The number and weight of graded components will needed by adjusted to compute your actual term grade).

James Tam

Calculating Your Overall Term Grade Point (2)

- You can use the spreadsheet on the course web page to estimate your term letter grade:
  - https://cspages.ucalgary.ca/~tam/2025/231P/2024P_231_grade_calculator.xlsx
  -

Contrast The Cut-Offs

Official UC cutoffs

Official university cut-offs	Letter	
4.3 and above	A+	4
4 to less than 4.3	A	4
3.7 to less than 4	A-	3.7
3.3 to less than 3.7	B+	3.3
3 to less than 3.3	B	3.0
2.7 to less than 3	B-	2.7
2.3 to less than 2.7	C+	2.3
2 to less than 2.3	C	2.0
1.7 to less than 2	C-	1.7
1.3 to less than 1.7	D+	1.3
1 to less than 1.3	D	1.0
0 to less than 1	F	0

The Tam cutoffs (this course)

Min GPA	Max GPA	Letter
0	Less than 0.85	F
0.85	Less than 1.15	D
1.15	Less than 1.5	D+
1.5	Less than 1.85	C-
1.85	Less than 2.15	C
2.15	Less than 2.5	C+
2.5	Less than 2.85	B-
2.85	Less than 3.15	B
3.15	Less than 3.5	B+
3.5	Less than 3.85	A-
3.85	Less than 4.1	A
4.1	Depends upon the semester	A+

- The cutoffs in the spreadsheet are **significantly more lenient** (almost everyone "gets a break" e.g. instead of 3.7 for an A- it's 3.5 (midpoint between A-/3.7 and B+/3.3 is the higher letter grade)
- Do not expect a further "rounding up" at the end of the term e.g. 3.4999999999999999999999999999999999999999999999999999 is soooooooooooooooooooooooooooooooooooooooooooooooooo close to 3.5 can't you get an "A-"
- No or using an Internet emphasis
- NOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO!! !!

Basic Computer Skills Assumed

- You know what a computer is!

- You've used a computer in some form (e.g., turn on, turn off, open a file, played a game, gone online etc.)

- You have experience *using common applications* (specifically email, web browsers, text editing using a word processor).

- That is, you already have basic computer literacy skills coming into this course.

What This Course Is About

- Writing/creating computer programs.

- But it is *not assumed* that you have prior knowledge of Computer Science (or even experience writing programs)

- It can be a lot of work.

Late night 'coding'

Satisfaction coming from solving that tough algorithm!

- The course can be completed by students with a normal course load (many already have gotten through it!)
- But be cautious if you already have many other commitments

Wav file from "Tam"

1st Year Programming Classes Are Challenging

- This fairly universal among accredited universities.
  - I have seen many outlines because I've evaluated transfer credits for other computer science courses from many other post secondary institutes.

- It's given that this course cannot be an exception.
  - Making the course easier than others would devalue it's worth Again: the majority of past students have gotten through it without past experience.

- Also: I've modified the course over time to help students develop the necessary skills.
  - But it's up to the students to follow those lessons and more importantly to practice those skills.
  - It's analogous to training under an athletic coach.

James Tam

How To Succeed

- How do successful people such as top: scientists, artists, athletes etc. "do it"?

- In these cases the are taught important things (such as scientific principles, artistic principles, body mechanics etc.)

- But also they 'do'…lots and lots of application/practice.

James Tam

How To Succeed In This Course

1. Practice things yourself (not by getting the answer from someone/someplace else).

Providing solutions to assignments may be popular among students but useless for learning (solves a single problem)

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

- 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.
  o At the *very least* attempt every assignment.
  o 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).
  o Write lots of little 'test' programs to help you understand and apply the concepts being taught.

- Trace lots of code (computer programs)
  o Involves reading through programs that other people have written, and executing it 'by hand' in order to understand how and why it works
  o This is an essential skill.
  o Relying on just running the program and observing the results won't always work (errors?)
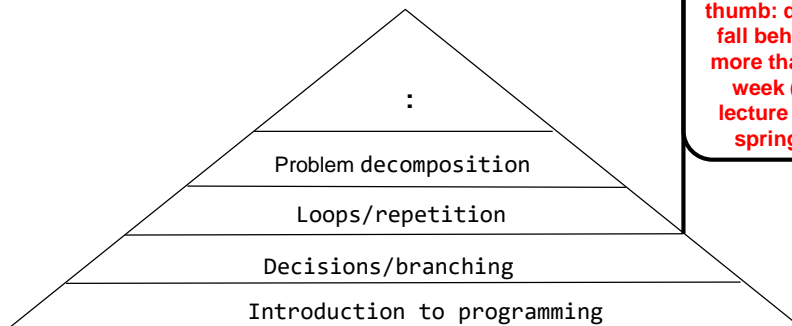- Extra practice problems for the course:
  o https://pages.cpsc.ucalgary.ca/~tamj/2025/231P/assignments/practice/index.html

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!

Rule of thumb: don't fall behind more than 1 week (1 lecture for spring)

```
          :

   Problem decomposition

   Loops/repetition

   Decisions/branching

   Introduction to programming
```

How To Succeed In This Course (4)

- If you find concepts unclear trying to research the answer on your own can be beneficial (because this is a 'hands on' field).
  - Trying to determine the answer on your own is of value.
  - Looking at online resources:
    - Keep in mind that some programming resources online just like other online information may not always be a good source.
    - Start with more reputable sources
      - www.ucalgary.ca/library
      - www.python.org
      - Writing and running test programs to test assertions.
- Addendum to the previous point #2 and a point raised earlier "ask questions".
  - Don't wait too long (more than a few days) to do this because latter concepts may strongly depend on your understanding of earlier concepts.

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 course notes.
   b) Look through the textbook(s)

When we get to more complicated programs that appear to 'jump around' in how they execute ("section: problem decomposition/functions") just having an idea of the scope and components of the program beforehand can be useful when I cover it in class.

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.
   - Some assignments may require the application of multiple concepts, not all the concepts have to be completely covered before you start working on an assignment.
     o Start working based on what's currently been covered (this will teach you how to decompose a program and work on it a part at a time).

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

James Tam