Principles of Computer Science

CMPT 145

Instructor Contact

- Michael Long
    - Office: Thorvaldson 373 (ask for Michael)
    - Email: `michae.long@usask.ca`
    - Office Hours: Spinks Lab (time TBA)

New First Year Computer Science Courses

We have created a sequence of three courses: CMPT 140, 141, and 145. It's a single stream of courses with two entry points, depending on your background.

CMPT 140 An introduction to programming for those with no prior programming experience.

CMPT 141 An introduction to programming in Python for those who have some programming experience.

CMPT 145 Builds on the basic programming and problem solving techniques introduced in CMPT 141.

Who should take CMPT 145?

Are you in the right place?

You should be ready for CMPT 145 if you have credit for:

- CMPT 141
- CMPT 115
- CMPT 111/113/116 + permission

Class Website

- All course content appears on class website at moodle.cs.usask.ca (must be registered in course to access).
- On course website you can find:
    - Course syllabus
    - Course announcements
    - Assignments, lecture and lab material, readings
    - Instructions about installing software on your personal computing machines
    - Video capture (hopefully)

Course Learning Objectives

- Build substantial applications for any field or discipline, making use of Python's extensive libraries.
- Implement computational solutions to a wide variety of problems, using a range of algorithmic strategies, and a range of data structures.
- Verify that Python programs work correctly.
- Assess and mitigate computational efficiency concerns that may arise in practice.
- To build good work/study/time management habits.
- Continue the formal study of computer science in courses such as CMPT 214, CMPT 260, and CMPT 270.
- See course syllabus for technical learning objectives.

Review of Syllabus

- Course syllabus is found on Moodle
- Learning objectives
- Grading scheme
- Midterm: in-class, June 21
- Lectures & Laboratories
- Assignments

Expectations

- Aim for 100% attendance in lecture.
- Aim for 100% attendance in laboratory.
- Aim for 100% completion of assignment work (not necessarily 100% grade)
- Reasonable allowances for absence, of course.
- Work in the Spinks labs, for availability to instructor's office hours, TA help desk hours, and your peers.

Managing Expectations: scheduling

Make a schedule, and stick to it.

- Schedule in your fixed commitments. All classes, work schedule, etc.
- Schedule in social time. Lunch breaks, video gaming, hanging out w/ friends, family, etc.
- For each hour of class, schedule 1-2 hours of study time before the class. Time to prepare for class (go over past notes, new readings, etc).
- Schedule in time to do all your homework/assignments. Indicate which homework to work on in each time slot

Note: You should end up scheduling 14-20 hours of productive time per week for a single course, not including lecture time.

Course Delivery

- Short pre-class readings
- In-class exercises, problem solving; solved as individual, small groups, class
- Weekly laboratories
- Weekly individual assignments
- Mid-term examination
- Final examination

Evaluation

(From the Course Syllabus)

10 assignments (4% ea.)	40%
10 lab exercises (1% ea.)	10%
Mid-term examination	10%
Final examination	40%

Important Dates

- Labs begin **May 14**
- Mid-term exam **June 21**, in-class, normal time and place.
- Final exam scheduled by registrar (April 15-17).
  Do not make travel plans until release of final exam schedule.

Important Policies

Details in Course Syllabus

- Ask lots of questions
- Don't be afraid of asking questions.
- No late assignment/lab submissions
  (sorry! sounds harsh, but we have a schedule to keep).
- Don't cheat.

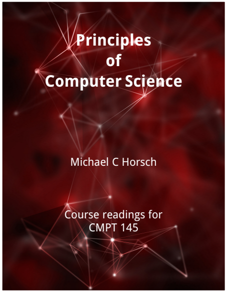
Important Policies

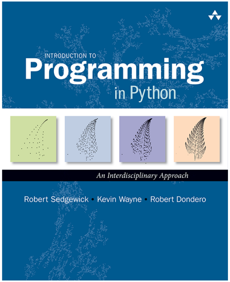
Read the Syllabus for detailed course policies.

It's your responsibility to read the syllabus.

Textbooks

- **Course readings** (FREE!): M. Horsch, Principles of Computer Science. Assigned readings from this book before each class.
- If you want a second opinion (**not** required): Sedgewick, Wayne, and Dondero. Introduction to Programming in Python. Addison-Wesley, 2015.




INTRODUCTION TO
Programming
in Python
An Interdisciplinary Approach
Robert Sedgewick · Kevin Wayne · Robert Dondero

Programming Language

- Python 3.5 or higher (**not** Python 2.x)
  Lab computers have the Anaconda Python distribution (works on Win/Linux/OSX)
- PyCharm IDE (students can get free one year academic license from jetbrains.com)
- We prefer you work on our lab computers.
- Guidance on how to install Anaconda Python and PyCharm on your own computer is on the website.
- We are not able to provide technical support for installation problems on your personal computing devices.
- Students jumping in without CMPT 141 should review the CMPT 141 readings (on the course Moodle).

Meet your Classmates

- Your turn! Get to know your neighbours.
- Turn to someone you **do not** already know.
- 3 minutes.
- Find out:
  - Neighbour’s name
  - Where they’re from
  - What program they’re in
  - Hobbies

Teaching Style

- We teach using an “Active Learning” technique.
- Assigned readings prior to each class.
  - You are responsible for reading and having a basic understanding of these before coming to class; in-class review will be minimal.
  - It is ~~okay~~ **fantastic** to come to class with questions!
- Interactive classes instead of speeches
  - Live demonstrations
  - A focus on individual and group exercises.
- Interrupt at anytime!
  - Ask questions early and often!

Supplies for Class

- Paper and writing instrument for notes and exercises.
- A notebook computer for class is entirely **optional**.
    - Don't buy one just for this class. Bring only if you already have one.
    - OSX, Windows, Linux, — it doesn't matter.