Introduction to Computing I

- ITI1120 Course Outline

Presented by: Mohammad Alja'Afreh , 05-09-2018

ITI1120 E: 2018

Professor: Mohammad Alja'Afreh
Office: CBY B505

- Course Web page: uottawa.brightspace.com (BRS)

- Office hours: to be determined
- Everything starts next week (check BRS)
- TAs: to be determined

(no such thing as "my TA")

Labs and Lectures

– Go to your assigned lectures

– Go to your assigned labs

- Read all the announcements carefully

Course webpage and online material

uottawa Brightspace:

- Supplemental course material (code from lectures, videos, some notes, labs ..)
- All communication via Brightspace
- Discussion forum (use it and subscribe to all topics)
- Assignments and Assignment Submission
- Grades

To access Brightspace go to uottawa.brightspace.com

Textbook

Main textbook (required):

"Introduction to Computing Using Python: An Application Development Focus, 2nd Edition" by Ljubomir Perkovic

- Paper version at uOttawa bookstore or order onine
- eBook online ($48.00)

https://www.wiley.com/en-ca/Introduction+to+Computing+Using+Python%3A+An+Application+Development+Focus%2C+2nd+Edition-p-9781118890943

Other recommended books

Another book: *Practical Programming (2nd ed): An Introduction to Computer Science Using Python 3"* by Paul Gries, Jennifer Campbell, Jason Montojo

The eBook is available as an eBook for $25.00.
http://pragprog.com/book/gwpy2/practical-programming

Free book: "*Think Python, How to Think Like a Computer Scientist*" by Allen Downey

Here are three Python 3 versions of the book:

- a "web" version here
  http://openbookproject.net/thinkcs/python/english3e/
- an interactive version (some material here is still in Python 2 instead of Python 3)
  https://runestone.academy/runestone/static/thinkcspy/index.html

Labs

- Scheduled labs will be in groups of about 30-40 students, with help available from a Teaching Assistant.
- You will be in a room with computers available for each student.
- All labs will involve programming in Python
- You should have been assigned to one of the available lab sections, and you must attend that session each week.
  - Your lab section assignment is available via ???
- The STE 0110 general lab is available at other times.

Labs

- Labs will start on Monday, 10th of Sept
- After that there are labs **every** week
- Labs are mandatory
- The first lab:
  - Assignment 0
  - How to use the Brightspace to submit assignments
  - Bring your own laptop if you had problems installing python. TA may be able to help

Assignments

- There will be up to 5 programming assignments during the term.
- All assignments will involve programming in Python; and maybe some will involve written work.
- Learn how to submit assignments.
- As test, submit Assignment 0 (in 1st lab).

Midterm Examination

- Duration: 1h20m

- Date: Saturday, November 3th
- Time: 5pm - 6:20 pm

- written exam, closed-book

Final Examination

- To be scheduled by the University Registrar
  - The exam date and time will be scheduled during the exam period
  - Check the university web site later for the date, time, and location.

- Written exam, closed-book.

Determination of Final Grade

- Labs (11 labs): 6%
  (0.6% each, 10 out of 11 labs)
- Assignments (5 assignments): ` 25%
- 1 Quiz: 5%
- Midterm: 24%
- Final examination: 40%

****

To pass the course one has to get 50% on the weighted average of all tests (i.e. quizzes, midterm and final). Obtaining less will result in failing the course. To be precise, the above grading scheme applies unless the formula below gives a number less than 50. If that is the case, the smaller of the above grading scheme and the below formula is the final grade.

(quiz_1_percentage * 5 + midterm_percentage * 24 + final_percentage * 40) / 69

Things to do right away

1. Check your lab schedule
2. Familiarize yourself with Brightspace (Virtual Campus)
3. Get the textbook
4. If you want to work on your own computer, install Python 3.7
5. Find out how to start IDLE, write your first python program. run it, save it, close it, open it again (in lab, at home …)

Course Policies

- Missed/late assignments: mark of zero.
  - One assignment may be exempted only if a the University of Ottawa Health Services certificate is given to the professor within one week of the due date.
  - In that case, the weight of missed assignment will be transferred to the final.

- Absence from midterm / final examination: see Engineering faculty regulations.
  - Illness requires The University of Ottawa Health Services certificate
  - In that case, the weight of the midterm will be transferred to the final.
  - Travel, employment, misreading the timetable are not valid excuses

Course Policies

- Evaluation scheme cannot be changed

Plagiarism

To reduce the need to worry about plagiarism, follow this rule of thumb:

Never look at any other person/group's assignment code, or have their code in your possession, in any portion or form whatsoever. Also never share your assignment code with other students.

Read the following three pages in detail

Academic Integrity

- What is academic fraud?
  - Misrepresenting someone else's work as your own:
    - Failure to cite sources, including the internet and discussions.
    - Use of the words of someone else without quotation marks or other highlighting.
  - Falsified lab data or citations.
  - Violation of examination regulations.
  - Tampering with academic evaluations.
  - Helping another student do any of the above.

Policy on plagiarism

Plagiarism is a kind of fraud: passing off someone else's work or ideas as your own in order to get a higher mark. Plagiarism is treated very seriously. The assignments you hand in must be your own and must not contain anyone else's ideas. Refer to the University of Ottawa's Policy on Academic Fraud for a more detailed description of plagiarism and sanctions.
http://web5.uottawa.ca/mcs-smc/academicintegrity/regulation.php

Guidelines to help avoid plagiarism

You may discuss assignments with friends and classmates, but only up to a point: you may discuss and compare general approaches and also how to get around particular difficulties, but you should not leave such a discussion with any written material. You should not look at another student's solution to an assignment on paper or on the computer screen, even in draft form. The actual coding of your programs, analysis of results, writing of reports and answering assignment questions must be done individually.

Downloading code or any other material from the Internet, and submitting it as your own work without credit is also plagiarism. If you do talk with anyone about an assignment, please state this in your assignment and state the extent of your discussion. If you use another resource (such as textbooks, internet resources, etc) when solving your assignment, include the proper reference.

Note that it is also a serious offense to help someone commit plagiarism. Do not lend your assignment answers, printouts, reports or diskettes, and do not let others copy or read them. To protect yourself against people copying your work without your knowledge, retain all of your old printouts and draft notes until the assignments have been graded and returned to you. If you suspect that someone has stolen a printout or diskette, contact your instructor immediately.

Helping Each Other

Although you must not solve your assignments with the help of others, there are still many ways in which students can help each other. For instance, you can go over difficult lecture or lab material, work through exercises, or help each other understand an assignment handout. This sort of course collaboration can be done in study groups or through the discussion group.

The Brightspace discussion forum can be used to discuss techniques and tools used in assignments, but the discussions should never mention or present any potential or partial solutions to assignment questions. You can consider creating a study group.

If in doubt about whether a question you are asking or answering is against these guidelines, ask your TA (teaching assistant) the question instead.

Detecting Plagiarism

Measures taken to detect plagiarism

1. TAs have been instructed to report any suspicious of plagiarism they find, when they mark assignments, to the professor.
2. Programs that you submit may be screened using plagiarism detection software that is very effective at detecting similarities. The professor will take appropriate measures, once plagiarism is detected on part or on the whole of an assignment. Note that copying or lending is considered to be equally serious offenses.

Reference

This document is a translation of the the French document prepared by Daniel Amyot and is based in part on "Policy on Plagiarism" by Dr. Amy Felty (Septembre 2002).

Faculty regulations (2018)

- **Mandatory withdrawal**, if **second failure** in a mandatory course
- Academic standing is evaluated after students have attempted 24 course credits
  - **Mandatory withdrawal**, if **CGPA** is below 3
  - **Probation**, if 3 ≤ **CGPA** < 5
- Note that the **passing grade** for first year courses is **D** (2), whereas 3 is D+, and **5** corresponds to **C+** (65-69%)

http://www.uottawa.ca/about/policies-and-regulations

General Suggestions

- Based on problems identified by the the Committee on Academic Standing
  - Stay engaged: attend classes, tutorials, etc.
  - Understand academic regulations
    - Do not take too many courses, 130 hours per course
    - Consider dropping a course in which you have low average
  - Take university seriously
  - Do not work too many hours (in part-time jobs)
  - Not making connect between mathematics and science to engineering and computer science applications

Suggestions for ITI1120

- Do not fall behind
- Program as much as possible (assignments, labs …)
- After class, run the code we did in class. See if you understand why it does what it does. Play with it.
- When programming, do not be afraid to make mistakes (make sure each small part of program is correct before moving to next part)
- Read a textbook and Practice, Practice and Practice
- Make use of office hours and TAs

Mentoring Centre

- Designed to **compliment** tutorials and office hours.
- Designed to teach **study skills** in the context certain courses.
- Helps students **meet other students** in the courses.
- **Workshops** are going to be provided on:
  - **Time Management**
  - **Stress and Anxiety**

ITI1120 Introduction to Computing I — Python

This course is about computational thinking, not about learning a programming language.

Python is a programming language that is easy to learn and very powerful.

- Used in many universities for introductory courses.
- Main language used for web programming at Google
- Widely used in scientific computation, for instance at NASA, by mathematicians and physicists
- Available on embedded platforms, for instance Nokia mobile phones
- Large portions of games (such as Civilization IV) are written in Python

Once you learnt programming in one language, it is relatively easy to learn another language, such as C++ or Java.

ITI1120 Introduction to Computing I

Python

Python programming language was developed by at the end of 80s by Dutch Programmer Guido van Rossum

In the Python community, he is known as a "Benevolent Dictator For Life", since he continues to oversee the Python development process.

Python is named after British sketch comedy from '70s Monty Python's Flying Circus

Python, is free and open-source software and has a community-based development model.

Python 2 vs. Python 3

There are currently two major versions of Python in use: Python 2 and Python 3.

Python 2, the latest version 2. 7 released in 2000

Python 3, a new version designed to rectify certain fundamental design flaws in the language. Unfortunately it is not backwards compatible

We will use Python 3.

. https://www.python.org/downloads/

More here for those interested:

https://wiki.python.org/moin/Python2orPython3

Why are you here?

"Every scientist and engineer must know some programming. It is part of basic education, like calculus, linear algebra, introductory physics and chemistry, or English"

Alan Perlis 1961

And more a modern version:

https://www.youtube.com/watch?v=nKIu9yen5nc

Why are you here?

Computer science is not computer programming. We teach programming to teach computational thinking:

- Solving problems (with a computer).
- Thinking on multiple levels of abstraction.
- Decompose a problem into smaller problems.
- A way of human thinking (not "thinking like a computer")
- Thinking about recipes (algorithms).
- 30 years ago the solution to a problem in science or engineering was usually a formula. Today it is usually an algorithm (DNA, proteins, chemical reactions, factory planning, logistics).

What is a program?

A program is a sequence of instructions that solves some problem (or achieves some effect).

For instance: Call your friend on the phone and give her instructions to find your favorite café. Or explain how to bake a cake.

Instructions are operations that the computer can already perform.

But we can define new instructions and raise the level of abstraction!

A program implements an algorithm (a recipe for solving a problem).