Dawson College  
Computer Science Department   
Course Outline

Introduction to Programming

**Course Number:** 420-BXC-03 **Ponderation:** 1.5-1.5-3  
**Credits**: 2 **Prerequisites:** None  
**Domain**: Language of Mathematics and Computer Science  
**Ensemble**: 2  
**Semester**: Fall 2019  
**Date Revised**: May 28, 2019

# Description

In this course, you will solve problems and express your creativity with computer programming. You will thus develop your capacity for *computational thinking*, a fundamental, everyday skill. Course problem sets are inspired by a variety of domains such as biology, music, finance, and gaming. By learning a programming language and understanding how software “apps” are created, you will be empowered to shape the technology around you and to take an active role in our information age.

Upon successful completion of this course, you will be able to:

* Use fundamental programming constructs such as decisions, repetition and functions
* Analyze a problem and abstract it into components or sub-problems
* Use a programming environment to implement software that solves the problem

# Statement of Competencies

This course fully addresses the following competency: “To use various mathematical or computer concepts, procedures and tools for common tasks.” (0012)

# Communication

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Teacher** | **Office** | **Contact** |
| 02 | Patricia Campbell | 3H.33 | MIO |

Teachers’ schedules will be posted outside their offices by the end of the first week of classes.

In addition to keeping up-to-date with any lecture content you might miss due to absence, you are expected to check Moodle regularly to keep track of course announcements and resources.

# Assessment

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Value** | **Duration (Approximate)** | **Date (Tentative)** |
| **Class Test 1** | 20% | 1 hour | Week 7 |
| **Class Test 2** | 30% | 1 hour | Week 13 |
| **Assignments** | 30% | n/a | Throughout the semester |
| **Project** | 20% | n/a | Last 3 weeks |

* Assignments handed in after the due date and time will be assigned a grade of 0 unless you have received an extension (as an accommodation of sudden illness or other exceptional circumstances) from your instructor.
* Assignments will not all have the same weight.
* Assessment activities are to be completed individually unless otherwise stated. The penalty for cheating and plagiarism is at the instructor’s discretion and may consist of assigning a grade of 0 for the assessment in question or for the entire course.
* Absence during scheduled assessments (tests) will result in grade of 0 for that assessment. If an absence can be justified, please contact your teacher no later than 1 week after the assessment date to make alternate arrangements as soon as possible.

# Course Methodology

The course consists of one 1.5-hour interactive lecture, one 1.5-hour lab session and 3 hours of homework per week. Outside of class, you will complete readings and practice exercises to prepare for lectures. In the lab and at home, you will perform hands-on tutorials and assignments to demonstrate your understanding of course concepts.

# Reference Materials and Equipment

Python For Everyone (Python 3) <https://books.trinket.io/pfe/>   
Think Python 2nd Edition (Python 3) <https://greenteapress.com/wp/think-python-2e/>

Students will be asked to create an account on [repl.it](https://repl.it/) to join the teacher’s classroom for labs and assignments.

All course materials are available through moodle.

*Both are free books. Think Python 2e is available under the* [*Creative Commons Attribution-NonCommercial 3.0 Unported License*](https://creativecommons.org/licenses/by-nc/3.0/)*, which means that you are free to copy, distribute, and modify it, as long as you attribute the work and don’t use it for commercial purposes. Python for Everyone and all of the teacher’s non-source code material is available under* [*Creative Commons Attribution-NonCommercial-ShareAlike License*](https://books.trinket.io/pfe/AB-copyright.html)

*All of the teacher’s source code examples are licensed under the* [*Apache License 2.0*](http://www.apache.org/licenses/LICENSE-2.0)

A USB flash drive and/or cloud storage (through services like Google Drive, OneDrive, Dropbox) is recommended to back up your coursework.

**Ministerial Objectives and Standards: Computer Literacy, Set 2**



# Tentative Schedule

Note that tentative assessment dates are listed in the “Assessment” section above. Concepts introduced early on are practiced throughout the semester. The second half of the semester allocates time to combining previously introduced concepts and exploring their nuances.

|  |  |
| --- | --- |
| **Week** | **Learning Objectives** |
|  | Introduction, what is programming   * Explain the role of software and hardware in a computing system * Contrast compiled and interpreted languages, program execution; development. * Explore programming as a sequence of instructions, using a limited instruction set (CUPS offline hands on Robot exercise) |
|  | The Elements of Python Programming   * What is Python / where is Python used * Programming a computer   Using data   * Variables and Identifiers * Declaring variables, assigning values * Variable naming, Python reserved words * Using literals and sample data: numbers ( int), strings * Use appropriate comments and identifiers to make programs readable. |
|  | More data   * Numeric data types: float and int * Python dynamic data typing * Apply the order of operations to evaluate arithmetic expressions * print() function decomposed * User input – User prompts, input() function |
|  | User data   * input function and strings * data conversion functions int(), float(), str() * arithmetic assignment operators |
|  | Comparisons for decision making   * Boolean logic * Comparison operators * Apply the order of operations to evaluate arithmetic expressions and boolean expressions. * if branching |
|  | String maniputlation   * String literals, recap * Operators with Strings (\* and +) * String functions, len(), upper, lstrip/rstrip/strip, concat, compare * Simple lists and dictionaries   Looping with strings, lists and dictionaires   * for loop |
|  | **Test1** |

|  |  |
| --- | --- |
|  | Repitition while loop   * border condition * zero iterations, infininte loops * simple conditions |
|  | Repitition while loop   * counter controlled loops * validation using loops * sentinel controlled loops |
|  | Using functions   * Structured programming * Contrasting built in functions with user functions * Coding a no argument, no return function * functions using arguments * function that return values   Using modules   * Python modules (sys, random, dates, timers) * Write our own |
|  | Continue with functions |
|  | Practice & Review   * Work on project * Possible additional topics as needed for students’ project and/or interest |
|  | **Test 2**  Work on project |
|  | Work on project |
|  | Work on project |