DAWSON COLLEGE
Science Program
COURSE OUTLINE

Introduction to Computer Programming in Engineering and Science

Course number	Ponderation	Hours
360-420-DW	3-2-3	75

Prerequisite

- Calculus II (201-NYB-05)
- Chemistry of Solutions (202-NYB-05)
- Waves, Optics and Modern Physics (203-NYC-05)

Corequisites

- Linear Algebra (201-NYC-05)

Note

Students who have successfully completed a complementary course with the course number 420-BWC or 420-BXCare not eligible to take this course.

Description

The Computer Science part of the course will introduce the student to the fundamentals of programming used to write applications that encompass building blocks (sequential, selection, and repetitive control structures) used in program construction. The student will learn how to analyze problems, and then design and implement both numerical and non-numerical (searching and sorting) algorithms to solve the problems.

The Physics part of the course will revisit material and problems that students will have already seen in previous science courses but will do so in a way that begins to consider their true complexity. Students will analyze complex problems, develop mathematical models and then solve the relevant equations using appropriate numerical methods with the aid of programming techniques developed in the Computer Science component. The Physics component focuses on modeling, and problems will be taken from a wide range of engineering and scientific fields.

Semester: Winter 2025 **Revision: 2024-12-19**

Computer Science

Teacher(s)	Office	Contact	Section
Nasreddine Hallam	3H.14	MIO	02
Lee Wilkins	TBA	MIO	01
Jeremie Choquette	7A.16	MIO	03

Physics

Teacher(s)	Office	Contact	Section
Jean-François Brière	7A.16	MIO	01, 03
Hélène Nadeau	7A.22	hnadeau@dawsoncollege.qc.ca	02

Course material (lectures, exercises, assignments) will be posted on Moodle. Students are expected to regularly

monitor this platform for updates/notifications.

Physics course material will be specified by the Physics course teacher(s).

Teachers' schedules of availability will be communicated online by the end of the first week of classes.

Teachers will respond to students' inquiries in a timely manner.

Course Objectives

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

- Solve problems using methods proper to science
- Demonstrate consistency and rigor in problem-solving
- Justify, orally and in writing, the approach used to solve the problem
- Apply a software development method to solve engineering and scientific problems
- Solve scientific problems that require the evaluation of mathematical equations
- Evaluate and use packages from software libraries to solve engineering and scientific problems
- Validate and assess the solution to the problem

This course will allow the student to partially achieve the competency 00UV: Apply the experimental method in a scientific field.

- Represent various situations, drawing upon relevant concepts, laws, and principles of science.
- Solve problems using a method proper to science.
- Apply techniques of experimentation or validation specific to science.

Course Methodology

Introduction to Computer Programming in Engineering and Science is an interdisciplinary course within the Science program, taught jointly by the Computer Science Department and the Physics Department.

The course consists of 75 lecture/lab hours of three periods a week. This will be given as two 1.5-hour sessions and one 2-hour session each week. All exercises and assignments will be drawn from both Computer Science and general sciences.

Of the 75 course hours, 45 will be taught by the Computer Science Department and 30 by the Physics Department.

Equipment and required software

Sufficient computer media (USB or cloud accounts such as OneDrive or Google Drive) to perform regular backups of course-related work.

All computers used on campus will have all the necessary software installed already. If the student wishes to use their own computer, they will need to have the following software installed on their computer:

- Python
- Visual Studio Code
- Jupyter notebook

Online readings

- Think Python (by Allen B. Downey, from Green Tea Press)
- How to Think Like a Computer Scientist (from Runestone Interactive tools)

Assessment of student performance

Computer Science Assessment Component (60%)

(60%)	Value	Date (tentative)
Test #1	15%	week 7
Test #2	15%	week 13

Three Assignments	30%	tentative

.

Physics Assessment Component (40%)

(40%)	Value	Date (tentative)
Assignments (4 x 2%)	8%	Date communicated by the Physics teacher
Project 1: Solving differential equations	10%	Week 11
Project 2: Applying programming in science	22%	Week 15

Tentative Schedule

The full details of the learning activities will be distributed to the students at the beginning of the course.

Physics – The Physics assignments and projects will start in week 6 and then they will continue until the end of the semester. The topics are chosen to reinvest the learnings from the computer science activities.

Computer Science - The Computer Science component will contribute towards all course objectives and competencies via the following learning activities

Week	Computer Science (CS) Theory	Assignment/Test
1	Understand basic Computer Science terminology and concepts: • Specify computer components: CPU, GPU, RAM, Drives. • Describe the nature and importance of operating systems. • Explain what a programming language is. • Briefly explain how apps are designed and executed. Intro to Python: • Write Python statements • Declare and initialize variables. Change values of variables. • Write arithmetic expressions • Invoke the print(), input() functions. • Supply functions arguments. • Trace program execution.	
2	Python basic syntax • Understand Python syntax: indentation, comments, syntax errors • Identify built-in data types (int, float, str, bool, etc.) • Distinguish between mutable and immutable variables. • Identify reserved words and built-in functions. • Perform mathematic calculations, named constants Code a Python program respecting naming conventions and guidelines. • Apply standard naming conventions. • Apply recommended indentation and program style techniques. • Apply recommended documentation techniques. Overview of development: • Explain the steps involved in the program development process (algorithm, flow chart, pseudo code) • Use Editor, IDE, and Notebook to code applications • Describe the purpose of libraries. • Differentiate between logic, syntax, and run-time errors. • Use functions to write modular/reusable code	
3/4	Solve and implement problems that require decision-making. • Explain the three basic control structures – sequence, selection	Assignment 1

	andrepetition. • Illustrate control structures using a flowchart. • Code simple if/else selection structures. • Code nested if/else selection structures. • Evaluate the efficiency of selection structures. • Code complex decisions. • Identify the order of precedence of relational and logical operators. • Understand short circuit evaluation. • Understand data validation and its importance to data integrity and system security. Working with strings • Use string functions • Comparing strings Solve problems that require the evaluation of mathematical equations. • Use arithmetic operators to construct expressions. • Understand the rules of precedence in an expression • Convert arithmetic expressions into Python expressions. • Evaluate Python arithmetic expressions. • Recognize the precision of numeric data types. • Use methods of the math class – e.g., pow(), sqrt(), random(), abs(), min(), and max().	
5/6	Solve and implement problems that require the repetition of a block of statements. • Use a for loop to perform counter-controlled repetition (use range) • Introduce list data types • Use a while loop to perform conditional repetition. • Determine the most appropriate repetitive structure. • Outline the functionalities of the break and continue statements. • Describe how and why to avoid the use of the break and continue statements. • Use loop nesting appropriately. Review for Test 1	
7-9	Design a function that solves a single identifiable task. Solve problems by constructing and using functions that perform one major task Define and invoke a function. Understand function scope, local and global scope Pass data values to a function. Distinguish between mutable / immutable parameters in function. Return values from a function. Define and use NumPy Create, manipulate arrays Use NumPy functions	Test 1 (week 7) Assignment 2 (week 8)
10	Declare and use built-in collections/containers: • Initialize a list, tuple, set, dictionary • Use a for loop to manipulate data stored in a container. • Understand and be able to use single and multiple dimension lists.	
11	Object-Oriented Programming	Assignment 3
12	Introduction to Data Analysis • Use Pandas series • Use Pandas series with dictionary • Use Pandas data frame • Read/save data in a file through Pandas	

	Introduction to Data Visualization • Use matplotlib • Adding title, legend, handling x axis and y axis	
13	Review for Test 2	Test2 (week13)

DAWSON COLLEGE
COMPUTER SCIENCE DEPARTMENT POLICIES

1. ACADEMIC INTEGRITY

The Computer Science Department adheres to the Dawson College Academic Conduct policy. Students have an obligation to inform themselves of all aspects of this policy. Every instance of cheating or plagiarism leading to a resolution that impacts a student's grade must be reported, with explanation, in writing, to the appropriate Dean. (ISEP Section V-A). The penalty for cheating or plagiarism may range from a penalty to zero on the work to a failure in the course.

2. ATTENDANCE AND LATENESS

Students are responsible for all material covered in classes and labs, whether or not they are present. Students have an obligation to arrive on time and to remain for the duration of scheduled classes, labs and activities. Students who disregard this obligation may be asked to leave the class, lab or activity. Students should refer to the Institutional Student Evaluation Policy (ISEP Section IV-C) regarding attendance.

3. RELIGIOUS HOLIDAY OBSERVANCE

Class period(s) may be cancelled in order that the teacher can observe religious holidays. Any material missed as a result, will be made up during labs, class periods and alternate tutorial sessions.
Students observing religious holidays must inform their teachers, in writing, as prescribed in the ISEP Policy on Religious Observances, no later than the end of the second week of the impacted semester or term. This applies both to the semester or term, as well as to any final examination period. (ISEP Section IV-D). Students absent from classes because of observance of religious holidays will not be penalized. It must be emphasized, however, that this College policy should not be interpreted to mean that a student can receive credit for work not performed. It is the student's responsibility to fulfill the requirements of the alternative arrangement.

4. SOFTWARE

The Computer Science Department forbids the use of the computer labs it uses to make any copies of any software without the explicit authorization of the Department. Apart from any legal action that might arise from such unauthorized copying, the Computer Science Department reserves the right to discipline any student involved in such activity.

Students who borrow software from the College and fail to return it will be placed on the defaulter list and be subject to the appropriate penalties.

Viruses are programs that attach themselves to a computer system without the permission of those to whom the system belongs. They are deliberately written to be, at worst, harmful (e.g. destroying the contents of disks) and, at best, bothersome (e.g. disturbing the image shown on the screen). They can cause serious losses of time and effort for students, staff and faculty. Any student involved in the deliberate spreading of viruses is subject to the most severe penalties prescribed by Dawson College regulations, apart from any legal action that might arise from such acts

5. LITERACY

The Computer Science Department recognizes that literacy in all its forms (read, written, spoken) is essential to students in their careers.

Teachers may choose to incorporate a literacy component into the marking scheme for any piece of work. Teachers may use their discretion to insist that any piece of work submitted for credit is revised by the student if it is unsatisfactory with regard to literacy.

Teachers will inform all students in their courses of this policy at the beginning of each semester either by including it in the course description or otherwise.

6. CELL PHONES

The use of cell phones in all Computer Science lectures, labs and exams is prohibited.

7. PORTABLE COMPUTERS

The use of portable computers in all Computer Science lectures for purposes other than note- taking is prohibited.

8. STUDENT CONDUCT

Everyone has the right to a safe and non-violent environment. Students are obliged to conduct themselves as stated in the Student Code of Conduct and in the ISEP section on the roles and responsibilities of students. (ISEP Section II - D)

9. PROFESSIONAL CONDUCT POLICY

Students who are enrolled in the Computer Science Technology careers program must conduct themselves according to the Professional Conduct Policy as described in the Program Handbook (ISEP Section IV-N.2). This professional conduct policy includes online conduct.

10. INTENSIVE COURSE CONFLICTS

If a student is attending an intensive course, the student must inform the teacher within the first two weeks of class of the specific dates of any anticipated absences.

11. ISEP

The Institutional Student Evaluation Policy (ISEP) is designed to promote equitable and effective evaluation of student learning and is therefore a crucial policy to read and understand. The policy describes the rights and

obligations of students, faculty, departments, programs, and the College administration with regard to evaluation in all your courses, including grade reviews and resolution of academic grievance. ISEP is available on the Dawson website. (link)