JOHN ABBOTT CEGEP/COLLEGE COURSE OUTLINE

A. GENERAL INFORMATION:

1. Program name(s) (if applicable):	**Science (200.B1)**
2. Course title:	Programming in Science
3. Course and section number(s):	420-SN1-RE - sections 1
4. Ponderation (weekly class – lab/fieldwork – homework hours):	1 - 2 - 3
5. Credits:	2.0
6. Competency statement(s) and code(s):	0F01: Develop software programs to automate problem-solving in a scientific context. Covered: Completely
7. Prerequisite (if any):	None
8. Semester:	Fall 2025
9. Teacher name(s), (pronouns if desired):	Elizabeth Poggie
10. Office number, phone extension (email address optional):	P-237
11. Teacher's availability:	Posted on LEA or Teacher's door

B. INTRODUCTION

Course summary:

This course introduces structured and disciplined approaches to computer programming and problem solving in a scientific context.

Students will gain the ability to analyze a word problem and design an algorithm to solve it. This includes implementation of the algorithm in `Python`, verification of correctness and accuracy of the algorithm, and displaying results either textually or graphically.

Role and place of the course:

Programming is a required skill for almost all scientific endeavours, whether to model data or analyze data or even predict outcomes using modern AI techniques. As a result, this course is offered in the first year of the science program, to ensure students can leverage their new skills in their future studies.

Other information (links to Integrative Activity, Comprehensive Assessment, Exit Profile, certificates…):

C. COURSE OBJECTIVES:

Upon successful completion of this course, students will be able to use the `Python` programming language to:

- take a word problem and design an algorithm to solve it
  - evaluate the data that is known
  - determine what data is unknown
  - determine the process required to calculate the data
- analyze an algorithm and its associated code to evaluate correctness, and fix any errors
- use appropriate libraries or IDEs to test the correctness of the algorithm/code
- obtain data from users and from structured documents
- display data using relevant plotting techniques
  - use effective sampling techniques and appropriate storage of data
  - apply filters to data to improve readability of the graphed data
- use standard science libraries such as `numpy`, `plotly`, `pandas` and others
- simulate physical models and statistical models
- implement standard algorithms in Python

Government Competency:

0F01 - Develop software programs to automate problem-solving in a scientific context.

Performance Criteria for the Competency as a Whole

- Appropriate use of terminology
- Correct use of required software development tools
- Observance of program development steps
- Respect for digital ethics regarding intellectual property
- Demonstration of autonomy, rigour and perseverance

	Elements of the Competency	**Performance Criteria**
0F01.1	Plan the automation for solving of a problem.	• Appropriate recognition of the concepts involved in a problem • Accurate determination of inputs and outputs • Accurate determination of the necessary processing • Coherent decomposition of the algorithm • Preparation of appropriate test cases to validate the functioning of the program

	Elements of the Competency	Performance Criteria
0F01.2	Code the algorithm in the programming language.	• Logical organization of instructions • Appropriate use of basic data types and arrays • Accurate use of arithmetic, relational and logical expressions • Adherence to programming language syntax and conventions • Appropriate use of libraries
0F01.3	Verify the program's operation.	• Correct tracing of program execution • Detection of operational errors • Relevance of the corrections made

D. EVALUATION PLAN

Evaluation type	%	Tentative date	Competencies/Objectives	[x] if part of final evaluation
Test	20%	Week 7-8	0F01.1, 0F01.2	
Final Exam	40%	After classes	0F01.1, 0F01.2	✓
Assignments (4-5)	30%	Distributed over the 15-week semester	0F01.1, 0F01.2	✓
Minor Assessments	10%	Distributed over the 15-week semester	0F01.1, 0F01.2, 0F01.3	✓
Total value:	*80%*	TBA	0F01.1, 0F01.2, 0F01.3	
Value of final eval (min 40%):	100%			

E. COURSE CONTENT:

Analyzing and Interpreting Problems

- From a problem statement, determine the Input/Output components (What is given/What is wanted)
- Design algorithms
- Write the pseudocode for an algorithm

Tools - PyCharm IDE - Trace and Debug - Online resources/documentation

Concepts required to implement solutions

- Basic Input/Output
- Data Types:
  - Numeric
  - String/Character
  - Boolean
  - Objects and calling methods
- Collections
  - Lists
  - Others such as ranges, tuples, sets and dictionaries
- Control structure:
  - Conditionals: if
  - Looping: using for
- Functions
  - Creating and using functions
  - Mutable and immutable arguments.
- Libraries
  - Importing and using external libraries.
- Graphing
  - Plot data

Quality of solutions

- Type hinting
- Commenting
- Documentation
- Naming conventions
- Testing
- Edge cases

Input / Output - Reading and writing to console and files. - Reading and writing basic data types. - Parsing structured data: delimited data.

Tentative schedule

Topics:	Week/Dates	Additional Info:
Files & Folders / Intro to the IDE and other required tools	1	
Basic Data types, variables & arithmetic operators	2	
Functions (& debugging) / Turtle Graphics	3	
Functions (& debugging) / IO functions	4	
Logical operators, Conditionals (& debugging)	5	
Lists / For loops	6	
Lists / For loops	7	
Review	8	Test
Algorithm Design & Implementation	10	
Graphing data	11	
Reading from a file	12	
Reading from a file	13	
Intro to `numpy` library	14	
Review	15	

F. REQUIRED TEXTBOOKS/MATERIALS, COURSE COSTS IN ADDITION TO TEXTS:

On-line resources, to be given as required

G. INSTRUCTIONAL METHODS:

Classes will be given in a computer science lab. The class time will be split between lecture and lab-time. Overall, the proportion of lecture to lab time will be one hour lecture for every two hours lab, but this split will differ daily depending on the current material being covered.

In class assessments can be done in a group or individually. Assignments must be individual work.

Assignments must be individual work.

I. PROGRAM, DEPARTMENTAL/DISCIPLINE, AND COURSE/SECTION POLICIES

Topic	Resource
Approved department attendance policy	Active learning of competencies requires hands on learning with interactive classroom work, which cannot be achieved with absences of more than 20%. Attendance is mandatory for the following program activities: • **Stage courses:** Absences that amount to more than 20% of the hours associated with the stage activity may mean that a student cannot achieve more than 50% on the stage course, resulting in the failure of the stage course. • **Stage evaluation meetings:** Absences that amount to more than 20% of the hours dedicated to these meetings will result in a grade of zero on this portion of the stage course, unless the absences have been deemed excused absences as per the IPESA. • **In class group work or project integration activities:** Absences that amount to more than 20% of the hours associated with any of these activities will result in a grade of zero on the assessment related to the missed activity, unless the absences have been deemed excused absences as per the IPESA.
Policy to ensure that issues relating to late submission, or resubmission, or work to be dealt with in an equitable manner	All assignments and projects are expected to be submitted by the required due date. A late penalty of between 5 to 10% per day might apply to assignments submitted late up to a maximum number of allowed late days. The exact percentage and the maximum will be indicated on the instructions specific to that assignment. Any work submitted after the maximum number of allowed late days has been reached will not be graded.

Topic	Resource
Policy dealing with the expectations of classroom behaviour, including use of cell phones, laptops and other technology	Not applicable.

J. COLLEGE POLICIES

Topic	Resource
Student rights and responsibilities (articles 3.2 and 3.3) Changes to evaluation plan in the course outline (article 5.3) Religious holidays (articles 3.2.13 and 4.1) Cheating and plagiarism (articles 9.1 and 9.2)	Policy 7: IPESA - Institutional Policy on the Evaluation of Student Achievement (version: 2021)
Cheating and plagiarism academic procedure and other resources	Academic Integrity: Cheating and Plagiarism Procedure (version: October 22, 2021) • **You need to log into Omnivox to access the above document** • **For PowerPoint on cheating and plagiarism** refer to the JAC Portal: My JAC Communities / Academic Council / Curriculum Validation Committee (CVC) / Course Outlines -- Reference Documents / Academic Integrity • For link to interactive tutorial on how to cite sources correctly: http://citeit.ccdmd.qc.ca
Code of conduct	Policy 13: Policy on Student Conduct and Discipline Procedures (version: September 21, 2021)