LaSalle College
Montréal

COURSE OUTLINE

Course identification

Name of program – Code:	COMPUTER SCIENCE TECHNOLOGY – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING – LEA.DQ
Course title:	**INTRODUCTION TO DATA STRUCTURES**
Course number:	420-A10-AS
Total number of course hours:	75 hours
Weighting:	2-3-3
Statement of the competencies – Codes:	Solve a problem – KP54 Develop programs – KP55

Contribution of the course in the program

Course position

This course is located in the second semester of *Computer Science Technology – Artificial Intelligence and Machine Learning* (LEA.DQ) program. Its duration is 75 hours divided into 30 hours of theory and 45 hours of exercises plus approximately 45 hours of homework. It shares the development of KP54 competency with *Design Patterns* (420-A13-AS) offered in in the third semester and *Algorithms and Data Structure* (420-A17-AS) offered in the fourth semester. Additionally, the current course shares the development of KP55 competency with *Introduction to Programming* (420-A03-AS) offered in the first semester, *Object-Oriented Programming* (420-A09-AS) offered in the second semester, and *Algorithms and Data Structure* (420-A17-AS) offered in the fourth semester. There are no prerequisites. Finally, the knowledge and skills developed in this course will be revisited in other courses of the program.

Scope of the course

During this course, students will learn three major principles, namely measuring the computational complexity of algorithms in terms of execution in runtime and memory management, implementing linear and nonlinear data structures, and finally working with variants of graph and tree data types. Such pillars enable students to develop optimal algorithms so as to reduce the load on the processing units such as CPU and GPU and consequently enhance the efficiency of the algorithms for real-time operations.

Upon completion of this course, students will be able to properly analyze computational properties of algorithms and potentially improve their execution performance according to the configuration of the available hardware (both processing units and memory). Additionally, they will have learned the use-cases of the linear and nonlinear data structures and their advantages over another for easy-to-complex problems. Since the programming language of the course is Python, learning techniques for improving the computational complexity of the algorithms directly contributes to implementing more efficient codes in this interpreter-based scripting language.

Course components (objective and standard of the competencies)

Expected outcomes (achievement context of the competencies)

The achievement context of these competencies will reflect the conditions as they occur in the following settings: academic, professional, work, or life environment.

Achievement context appropriate to the competency – Problem Solving – KP54

- Based on a problem.
- When performing a task, solving a problem, or making a decision.
- Based on the needs of everyday life.
- Using reference documents (written or other).
- Based on values, the organizational culture, company policies and rules.
- Using decision mapping.
- Based on a computerized information management system (GIS), appropriate software
- applications and electronic platforms.
- Using advances in new information and communications technologies (NICT).
- Using relevant description and visualization methods (e.g. class diagram)
- Based on actual and synthesized data
- Using quantitative and qualitative approaches (where applicable)

Achievement context appropriate to the competency – Program development – KP55

- In different types of work environments
- Based on predefined tasks or algorithm
- Based on a data model
- Using appropriate software and reference documents
- Respecting the license agreement

Throughout the course, you will engage in various learning situations/activities so that by the end of the course, you will have met the expected outcomes.

Elements and performance criteria

The elements of an objective formulated in terms of the competency specify its essential components. They include only what is necessary in order to understand and master the competency. If the competency is described as a process, the elements are the steps for execution.

The performance criteria are the specific pre-established requirements upon which you and your teacher can objectively judge your development of the targeted competency. They are part of the description of this competency. They are prescriptive.

Sometimes an element appears in more than one course. If this is the case, a number indicates its complexity level: level one (1) being the simplest, level two (2), average, and level three (3), advanced, at the ministerial level.

Below are the elements of the competencies and performance criteria for this course that are to be respected:

Competency: Program development – KP54

General ministerial and institutional performance criteria:

- Autonomy
- Initiative
- Analytical mind and critical thinking

Elements of the competency	***Performance criteria specific to each element***
1. Define the problem	1.1 Accurate identification of the system under study 1.2 Relevant collection of internal and external information sources 1.3 Accurate visualization of the known(s) and the desired unknown(s) 1.4 Effective use of appropriate concepts, units, symbols, and language 1.5 Accurate identification of constraints with respect to the individual, collective, spatial, temporal, legal, and cultural dimensions of the problem 1.6 Relevant establishment of the importance of different factors and the criteria for success Proper formulation of the problem statement
2. Interpret the causes	2.1 Appropriate consideration of psychological and social factors characterizing the people involved 2.2 Appropriate analysis of important elements and challenges 2.3 Reliable assessment of the scope, effects, and the required knowledge, materials, tools, and resources 2.4 Complete collection of relevant information 2.5 Appropriate breakdown of the problem into smaller/sub-problems
3. Plan a solution	3.1 Proper identification of common and creative problem-solving strategies 3.2 Relevant detection of general processes and repeated patterns 3.3 Clear description of the application logic and interface to generate or program

	3.4 Effective application of the appropriate problem-solving technique to subproblems 3.5 Realistic use of reflection, imagination, and innovation to search for ideas 3.6 Proper assembly of solutions to the subproblems
4. Developing the solution design	4.1 Accurate graphical representation of the different models 4.2 Accurate drafting of unit, integration, functional or acceptance test plans 4.3 Modelling of a database or object-oriented design aligned with user needs 4.4 Clear description of the application logic and the interface to be generated or programmed 4.5 Accurate graphical representation of the different models 4.6 Active participation in the design review 4.7 Compliance with application development standards, methods and best practices
5. Evaluate the solution	5.1 Accurate assessment of soundness, efficiency, and feasibility of the planed solution 5.2 Adequate consideration of the participants' reactions to decisions 5.3 Methodical development of critical thinking 5.4 Relevant description of installations.
6. Carry out the plan	6.1 Correct implementation of the solutions to the subproblems 6.2 Proper refinement of ideas according to the solution evaluation where applicable 6.3 Concise presentation of the elements necessary to solve the problem 6.4 Accurate implementation of the final solution to the problem

Competency: Program development – KP55

General ministerial and institutional performance criteria:

- Critical thinking, methodical, analytic and synthetic.
- Naming conventions.
- Autonomy
- Sense of organization

Elements of the competency	***Performance criteria specific to each element***
1. Analyze the application development project.	1.1 Accurate analysis of design documents 1.2 Selection of data relevant to the application 1.3 Proper identification of the tasks
2. Organise the data structure.	2.1 Proper analysis of the data model 2.2 Accurate restructuration of the data model.

3. Prepare the computer development environment.	3.1 Proper installation of development environment and database management system 3.2 Proper installation of libraries or tools 3.3 Appropriate configuration of the version control system 3.4 Proper importing of the source code
4. Write the program.	4.1 Proper use of the libraries 4.2 Appropriate programming of modules 4.3 Optimal use of the programming language
7. Optimise the program.	7.1 Appropriate setting of program performance 7.2 Appropriate interpretation of results 7.3 Optimal functioning of the program
8. Produce the documentation.	8.1 Complete identification of information 8.2 Comprehensive API documentation 8.3 Accurate citation of code 8.4 Effective integration of an accessible help section

Course content/main themes

*Listed below is the **essential** content to be covered in this course:*

- Review basics of Python programming
- Computational complexity in terms of executionin in runtime
- Computational complexity in terms of memory management
- Linear and nonlinear data structures
- Graph and Tree data types
- Dynamic programming
- Traversing algorithms and their applications
- Search and sort algorithms
- Abstract data types

Learning activities

Provided below are examples of learning activities that correspond to the competencies for this course. The learning activities are found in the course calendar that complements this course outline.

- Lab exercises for all the four principles as mentioned above.
- In-class demonstrations by teacher and students
- Delivering multiple projects within the time frame of four weeks during a semester
- Implementing group projects to practice collaboration and leadership skills

Terms for Evaluating Learning

The evaluation of your learning is based on two inseparable methods: formative evaluation and summative evaluation. These two evaluation types are formal. Detailed information on the evaluation schedule is found in the course calendar, under the "Formative and summative evaluation schedule" column.

Formative evaluation

Following a learning activity or learning period, time is set aside for introspection. You will determine what has been understood and achieved and seek to identify the nature and origin of weak areas. These designated periods consist of simple means: short tests, association games, logbooks, a portfolio, questions, creating of samples, etc.

Formative evaluation is frequent and covers as many aspects as possible. It takes place in class, individually or in groups, and leads to immediate decisions. ***You are the one who assumes the bulk of the work during individual or group corrections, adjustments and other self-evaluation tasks. The purpose is not to determine grades.***

If you take the results of the formative evaluations seriously throughout the course, you will ensure preparedness for the summative evaluations. You will be able to make the necessary progress to acquire the targeted competency at the required level, according to the achievement context and pre-established performance criteria.

Below are some examples of formative evaluation methods that correspond to the targeted competencies for this course:

- In-class demonstration for all the sessions
- Industry-oriented assignments according to the needs and trends of the market
- Group projects for practice brainstorming and coming up with new ideas
- Constant discussions with teacher to get weekly feedback about student progress

Summative evaluation

Summative evaluations are less frequent. They take place later on, towards the middle and end of the semester. This gives you the time to integrate your learning and to learn how to apply it to situations related to the targeted competency. The summative evaluation material is prepared by your teacher according to the description of the course's targeted competency: its elements, achievement context and performance criteria.

The work completed in summative evaluations is graded. The purpose is to determine what you have learned.

Below is the information on the summative evaluation schedule and details for this course, as well as the weighting of marks:

Evaluations	***Weighting***
Midterm exam	30%
Project	30%
Final exam	40%
Total	100%

Institutional requirements

Student’s commitment

By registering for this course, you commit to:

- *obtain the necessary course materials at the start of the semester;*
- *respect the copyright;*
- *participate in the learning activities, formative and summative evaluation activities outlined in the course calendar;*
- *complete the work assigned to you;*
- *submit the work on time.*

Teacher’s commitment

Your teacher commits to:

- *create varied learning situations that enable you to put into practice the knowledge, actions and professional behaviour of the targeted competency;*
- *plan sufficient and appropriate formative evaluation activities, involving correction and improvement, that provide frequent feedback, allowing you to be well informed of your progress;*
- *provide summative evaluations that correspond to the course’s targeted competency;*
- *evaluate work according to the applicable criteria, in a fair and equitable manner within a reasonable time.*

The Institutional Policy on Evaluating Learning (IPEL) is applied to all institutional programs. Listed below are a few of its clauses:

Written language *(article 5.7)*

The teacher is responsible for identifying spelling and grammar errors and for allocating the corresponding number of marks for any given summative evaluation.

Below is the % – based on language requirements – that can be attributed to each summative evaluation:

- up to 5 %

Class attendance *(article 5.12)*

Attendance and participation in classes and evaluations are mandatory for all students.

The teacher has the responsibility of monitoring attendance and of evaluating the reasons justifying student absences from classes.

A student whose absences exceed the allowable number for the course could be denied access to the final exam for that course.

Plagiarism and fraud *(article 5.16)*

Plagiarism, attempted plagiarism or complicity in plagiarism during an assignment or any evaluated task contravenes the rules. This includes (but is not limited to):

- *the whole or partial presentation (reference, paraphrase, summary, translation, insertion) of the work of another (text, illustration, film, music, etc. on paper or online) as one's own, or failing to cite a source;*
- *the use of another student's exam during an exam;*
- *the use of an assignment done for another course or a project already submitted in the past, which is passed off as an original work.*

Fraud, attempted fraud or complicity in fraud constitutes an infraction.
This includes (but is not limited to):

- *the possession or use of any unauthorized document, material or equipment during an exam, including the use of technological tools;*
- *the execution of an evaluated task by another person;*
- *the substitution for another person during an exam, assignment or any evaluated task;*
- *the possession of the questions or answers of the exam;*
- *the obtainment of any aid not authorized in advance by the teacher.*

Plagiarism, attempts at plagiarism or fraud, or collaboration in plagiarism or fraud are prohibited and considered serious offences. Thus, any instances of plagiarism or fraud will lead to a grade of '0' for the assignment in question. In addition, a note will be made in the student's file and the student will receive a written notice from his or her Program Directorate to that effect.

In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. A second note is made in the student's file and the student will receive a summons from his or her Program Directorate. For a third offence, he or she may be expelled from the College.

Submission of work and tests *(article 5.8)*

All assignments must be submitted in class at the time designated by the teacher. Any late submissions result in a grade of zero (0).

Upon presentation of an official supporting document or valid reason for the absence, the student may request an extension from the teacher, who may accept or refuse the student's work and apply a penalty for the lateness.

Program Directorates do not accept student work. Assignments must be submitted directly to the teacher.

Rules and regulations to follow

Late arrivals

The teacher may refuse to admit to the classroom any student arriving late. A late arrival is considered an absence for that period.

Note: Students arriving late must recognize that the information they missed will not be repeated. Late students are therefore responsible for asking their peers about the material they missed. Arriving after the break, as well as leaving before the end of the class, may result in one or more hours of absence.

Eating and drinking in class

Eating and drinking are prohibited in the classrooms, locker rooms and Documentation Centre. Food may only be eaten in the cafeteria, vending machine areas and student lounges.

Mandatory course material

- Laptop with specifications mentioned on the college's website

LaSalle College. Bring Your Own Device. 2017. http://www.lasallecollege.com/futur-students/bring- your-own-device

Bibliography for this course

- GOLDWASSER, Michael H., GOODRICH, Michael T., TAMASSIA, Roberto. *Data Structures and Algorithms in Python*. Wiley, 2013.
- LEE, Kent D. and HUBBARD, Steve. *Data Structures and Algorithms with Python*. Springer, 2015.

Academic Studies Directorate approval:

Emmanouela
Tisizi pour la DÉ

Signature numérique de
Emmanouela Tisizi pour la DÉ
Date : 2024.09.18 11:52:43
-04'00'