LaSalle College
Montréal

COURSE OUTLINE

Course identification

Name of program – Code:	COMPUTER SCIENCE TECHNOLOGY– PROGRAMMING - 420.BP
Course title:	**DATA STRUCTURES**
Course number:	420-SD4-AS
Total number of course hours:	60 hours
Weighting:	2-2-3
Statement of the competency– Code:	Develop native applications with a database -00SS

Contribution of the course in the program

Course position

This course is located in the fourth session of the *Computer Science Technology- Programming* (420.BP) program. Its duration is 60 hours divided into 30 hours of theory and 30 hours of exercises (computer lab work) plus approximately 45 hours of homework. It finalizes and shares the development of 00SS competency with *Multi-tier Applications Development* (420-DA3-AS*)* given in the third semester.

Algorithms and Programming (420-AP1-AS) offered in the first semester is a prerequisite for this course. Finally, even if it is not a prerequisite to any other course, the knowledge and skills developed in this course will be revisited in *Workplace Integration: Programming (420-EP6-AS)* in the sixth semester.

Scope of the course

During this course, it focuses on providing students with strong knowledge and skills related to data structures such as linked lists, stacks, queues, trees including binary trees. Advanced searching and sorting algorithms are also discussed to help students in solving complex programming problems.

Upon completion of this course, students will have a solid understanding of data structures and algorithms, both the theory and the implementation. Students are able to design and implement proper data structures and efficient algorithms for different types of application development such as mobile and web applications.

Course components (objective and standard of the competency)

Expected outcomes (achievement context of the competency)

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

Develop native applications with a database -00SS

- For different target platforms: tablets, smartphones, desktop computers, etc.
- For new applications and applications to be modified
- Based on design documents
- Using a compiler designed for the target platform, a cross compiler or an interpreter
- Using an emulator on the development platform
- Using images
- Using issue tracking and version control procedures

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 competency and performance criteria for this course that are to be respected:

Competency:

Develop native applications with a database -00SS

General ministerial and institutional performance criteria:

- Critical thinking;
- Methodical, analytic and synthetic mind;
- Programming efficiency;
- Autonomy;
- Initiative.

Elements of the competency	***Performance criteria specific to each element***
5. Program the application logic.	5.1 Proper programming or integration of authentication and authorization mechanisms 5.2 Proper programming of interactions between the graphical user interface and the user 5.3 Appropriate choice of clauses, operators, commands or parameters in database queries

	5.4 Correct handling of database data 5.5 Proper programming of data synchronization 5.6 Appropriate use of data exchange services 5.7 Proper application of internationalization techniques 5.8 Precise application of secure programming techniques
6. Control the quality of the application.	6.1 Precise application of test plans in the emulator and on the target platform 6.2 Thorough reviews of code and security 6.3 Relevance of the corrective actions 6.4 Compliance with issue tracking and version control procedures 6.5 Compliance with the design documents
7. Participate in the deployment of the application.	7.1 Appropriate preparation of the application in view of its deployment or installation 7.2 Proper deployment or installation of the application
8. Produce the documentation.	8.1 Proper identification of the information to be written up 8.2 Clear record of the work carried out

Course content/main themes

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

Chapter 0: Introduction to Data Structures

- Abstract Data Types (ADTs).
- Definition of a Data Structure.
- Types of Data Structures: Linear Data Structures and Non-Linear Data Structures.

Chapter 1: Performance Analysis and Big-O Notation

- Big-O Notation.
- Implementation and analysis of sort algorithms : Bubble sort, Insertion sort, and Selection sort.

Chapter 2: Recursion

- Binary search in an array.
- Search the minimum/ maximum element in an array.
- Merge sort and Quick sort.

Chapter 3: Stacks and Queues

- Definition of a Stack (LIFO) and its operations.
- Array-based Implementation of a Stack.
- Definition of a Linear Queue (FIFO) and its operations.
- Array-based Implementation of a Linear Queue.
- Definition of a Circular Queue and its operations.
- Array-based Implementation of a Circular Queue.
- Definition of a Priority Queue and its operations.
- Array-based Implementation of a Priority Queue.
- Linked List-based Implementation of a Stack.
- Linked List-based Implementation of a Queue.

Chapter 4: Singly Linked Lists

- Introduction to Singly Linked Lists.
- Basic Operations on Singly Linked List (Insertion; Searching; Deletion; Traversing the list; Traversing the list in reverse order using recursion).
- Pointer-based Implementation of a Singly Linked List.
- Complexity Analysis (Big -O) of Singly Linked List operations.

Chapter 5: Doubly Linked Lists

- Introduction to Doubly Linked Lists.
- Basic Operations on Doubly Linked List (Insertion; Searching; Deletion; Traversing the list; Traversing the list in reverse order).
- Pointer-based Implementation of a Doubly Linked List.
- Complexity Analysis (Big -O) of Doubly Linked List operations.

Chapter 6: Binary Search Trees

- Introduction to Trees
- Binary Trees; Binary Search Trees (definitions and operations).
- Traversing a Binary Search Tree (Pre-order; In-order; Post-order; Breadth First).

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.

- Application exercises following teacher's demonstrations
- Practical problem-solving exercises
- Learning by project
- Case study
- Oral presentation of the final project

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:

- Homework assignments and labs
- Group discussions
- Scenario exercises
- Case study
- Teacher's feedback following students ' application exercises

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***
Mid-term Exam	30 %
Final 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 10%

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

- USB Key or cloud account for saving work.
- 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

- Michael McMillan. *Data Structures and Algorithms Using C#. First Edition.* Cambridge University Press., 2007.
- Michael T. Goodrich and Roberto Tamassia. *Data Structures and Algorithms in Java. Fifth Edition. Wiley Edition*., 2010.
- Yang Hu. *Easy Learning Data Structures & Algorithms C#: Data Structures and Algorithms Guide in C#.* 2019.

Academic Studies Directorate approval: *Signature and date of approval*

Daniel Dubreuil pour la DÉ

Signature numérique de Daniel Dubreuil pour la DÉ
Date : 2022.06.06 16:53:51 -04'00'