Course Outline

Full-Stack Developer – LEA.BN

A. General Information

Course title	**Data Structures and Algorithms**
Course number	420-JC3-AB
Hours	45
Ponderation *Ratio of lecture, practical and homework hours*	1-2-3
Credits	2.00
Competency statement(s) and code(s)	00SS - Develop native applications with a database Elements 5 & 6 only: 00SS.5 Program the application logic 00SS.6 Control the quality of the application
Prerequisite (s)	420-JB4-AB Programming II, 420-SA5-AB Database
Cohort	FSD-05
Start date	November 8, 2022
End date	November 18, 2022
Day(s) and times	M-F: 9:00-12:00 & 12:30-2:30
Classroom/lab number	Online
Semester	F2022
Teacher	Pargol Poshtareh
Teachers' contact info	
Course format (F2F, online, hybrid)	

B. Introduction

This course is part of the Full-Stack Developer program leading to an Attestation of Collegial Studies (A.E.C.). It should be taken in the third semester of the program.

In this course, students will develop their capacity to interpret, select and adapt algorithms to solve problems in a given situation. They will begin by learning about the importance of data structures in order to manage large amounts of data efficiently. The student will then apply data structures to produce efficient algorithms. The course will include topics such as stacks, queues, lists, and trees. There will be an emphasis on cultivating an attitude and approach to problem solving, testing, and code review.

C. Course Objectives

By the end of this course, students should be able to perform the following:

	00SS
Statement of the Competency	**Achievement Context**
Develop native applications with a database.	• 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
Elements of the Competency	**Performance Criteria**
5. Program the application logic.	• Proper programming or integration of authentication and authorization mechanisms • Proper programming of interactions between the graphical user interface and the user • Appropriate choice of clauses, operators, commands or parameters in database queries • Correct handling of database data • Proper programming of data synchronization • Appropriate use of data exchanges services • Proper application of internationalization techniques • Precise application of secure programming techniques
6. Control the quality of the application.	• Precise application of test plans in the emulator and on the target platform • Thorough reviews of code and security • Relevance of the corrective actions • Compliance with issue tracking and version control procedures • Compliance with the design documents

D. Evaluation Plan

Evaluation task	%	Approximate date	Link to competency(ies) and element(s)	Select if part of the final evaluation!
Class exercises (2 @ 5%)	10		00SS.5-00SS.6	☐
Assignment 1 (Unit Testing)	10	Class 3	00SS.5-00SS.6	☐
Mid-Term Exam	30	Class 5	00SS.5-00SS.6	☐
Assignment 2 (Code Review)	10	Class 7	00SS.5-00SS.6	☒
Final Exam	40	Class 9	00SS.5-00SS.6	☒

E. Course Content and Schedule

Course Content

Arrays, Lists
- multidimensional arrays, jagged arrays,
- lists of lists, lists of objects with lists
- caching: saving a value, reusing, invalidating
- encapsulation of lists and other data structures (never give away the reference)

Lists, stacks, queues
- lists, stacks, queues, priority queues (stack machine language example) HashMap, LinkedHashMap, TreeMap, HashSet, LinkedHashSet, TreeSet

Sorting, unit testing
- singly linked list, doubly linked list
- unit testing
- bubble sort algorithm example
- efficiency of an algorithm:
  - execution time O(1), O(log(n)), O(n), O(n^2)
  - planning and profiling memory use
  - how to use a profiler to track execution time

Recursion
- tail recursion, fractals, directory scanning,
- graphs and trees - their implementation and use

Design patterns
- May include:
- factory pattern
- singleton pattern
- visitor pattern
- iterator design pattern
- observer pattern
- functional strategy pattern
- service callback pattern

Schedule

Date or class	Topic(s)	Additional info	F2F	Online
Class 1	Arrays, Lists		☐	☐
Class 2	Lists, stacks, queues		☐	☐
Class 3	Sorting, unit testing, **Assignment 1**		☐	☐
Class 4	Sorting		☐	☐
Class 5	**Mid-term**		☐	☐
Class 6	Recursion		☐	☐
Class 7	Design patterns, **Assignment 2**		☐	☐
Class 8	Design patterns		☐	☐
Class 9	**Final Exam**		☐	☐

F. Required Textbooks / Materials / Costs

Title / Item	Cost $
N/A	0
Technical requirements for this course (hardware, software, High speed Internet connection, etc.)	

G. Bibliography (books, articles, videos, websites, podcasts, etc.)

Optional:

- Deitel, P. J., & Deitel, H. M. (2018). *Java: How to program early objects* (11th ed.). New York, NY: Pearson. ISBN-13: 9780134743356.
- Liang, Y. D. (2020). *Introduction to Java Programming and Data Structures: Comprehensive version* (12th ed.). New York: Pearson. ISBN-10: 0136520235, ISBN-13: 9780136520238.
- Sierra, K., & Bates, B. (2005). *Head First Java* (2nd ed.). O'Reilly Media Inc. ISBN-10: 0596009208, ISBN-13: 9780596009205.

H. Teaching Methods

The course is a combination of theory and practical work.

Students will be required to:

- Work alone
- Work in groups

It requires your individual presence and your active, consistent and sustained participation in your individual work. Your individual responsibilities are to complete the work assigned and be ready to work at the start of each class.

Hands on experience is mandatory to your success in this course.

Léa, the course management system within Omnivox, will be used in this course.

Learning Activities:

- Lectures/Demonstrations.
- Hands-On Exercises/Assignments: Case problems, concepts reviews, skills practice, and code review will help support and reinforce material in the course. These will be structured to be as realistic as possible given the time available.
- Tests
- Classroom Activity: Participation and Discussion

I. Departmental Policies and Classroom Policies

Classroom Policies

Late submission of work Work submitted late will result in a 10% deduction from the grade, per calendar day
Classroom behaviour
Online etiquette

Departmental Policies

Please refer to the following document concerning policies in place at the Centre for Continuing Education:

Continuing Education Policies and Guidelines
(version: December 1, 2020)

A. College Policies

Please refer to the following document concerning the provisos related to course outlines as a response to Covid-19.

Provisos for Course Outlines (Covid-19)
(version: winter 2022)

Topic	Resource
Student rights and responsibilities (see articles 3.2 and 3.3)	Policy 7:IPESA - Institutional Policy on the Evaluation of Student Achievement (version: June 12, 2019)
Changes to evaluation plan in the course outline (see article 5.3)	
Religious holidays (see article 4.1)	
Cheating and plagiarism (articles 9.1 and 9.2)	
Cheating and plagiarism	Academic Integrity: Cheating and Plagiarism Procedure (version: October 22, 2021) ***You will need to log into Omnivox to access this document.***
Code of conduct	Policy 13: Policy on Student Conduct and Discipline Procedures (version: September 21, 2021)

DISCLAIMER: Policies may be updated during the academic year. Should a link in the section above no longer work, please refer to the college website: https://www.johnabbott.qc.ca/the-college/official-documents/