Course Outline

Full-Stack Developer – LEA.BN

A. General Information

Course title	**Programming I**
Course number	420-JA4-AB
Hours	60
Ponderation *Ratio of lecture, practical and homework hours*	2-2-3
Credits	2.33
Competency statement(s) and code(s)	00Q2 - Use programming languages
Prerequisite (s)	None
Cohort	FSD-10
Start date	Jun 7, 2023
End date	Jun 22, 2023
Day(s) and times	M-F: 9:00-12:00 & 12:30-2:30
Classroom/lab number	
Semester	S2023
Teacher	Pargol Poshtareh
Teachers' contact info	MIO
Course format (F2F, online, hybrid)	Online

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 first semester of the program.

This course begins with an introduction to flow charts used to model simple algorithms and decision-making structures. These are then implemented using the Java programming language, where the student learns about program structure, variable types, passing parameters, control structures (if/else, switch, do-while, while, for) as well as the creation and use of subroutines (methods) and arrays. The student also gains a basic understanding of computer program structure, algorithm design, debugging strategies, unit testing, and source control.

C. Course Objectives

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

00Q2	
Statement of the Competency	**Achievement Context**
Use programming languages.	• For problems that are easily solved • Using basic algorithms • Using a debugger and a functional test plan
Elements of the Competency	**Performance Criteria**
1. Analyze the problem.	• Correct breakdown of the problem • Proper identification of input and output data and of the nature of the processes • Appropriate choice and adaptation of the algorithm
2. Proper identification of input and output data and of the nature of the processes	• Appropriate choice of instructions and types of elementary data • Efficient modularization of code • Logical organization of instructions • Compliance with the language syntax • Computer code consistent with the algorithm
3. Appropriate choice and adaptation of the algorithm	• Efficient use of the debugger • Identification of all errors • Astute choice of debugging strategies • Relevance of the corrective actions • Clear record of solutions to the problems encountered
4. Translate the algorithm into a programming language.	• Attitudes and behaviours that demonstrate thoroughness • Identification of all operational errors • Relevance of the corrective actions • Proper functioning of the program • Clear record of information concerning tests and their results

D. Evaluation Plan

INSTRUCTIONS FOR TEACHERS CONCERNING FINAL EVALUATIONS

40% MINIMUM For the final evaluation	**40% MAXIMUM** For a single evaluation task
Each course must have some form of final evaluation of sufficient weighting to attest the student's achievement of the competencies and the competency elements attached to the course. This evaluation should account for a minimum of 40%.	Given the intensive nature of the Continuing Education A.E.C. courses, a single evaluation task may have a maximum weight of 40% of the final grade. In other words, a single evaluation task with a weighting of 41% and above is not recommended.

The final evaluation may include several evaluations tasks. When combined, these may exceed 40%.

Examples of a final evaluation
1 final evaluation with a weight of 40%.
5 evaluation tasks with a weight of 10%, for a total of 50%.
1 evaluation task with a weight of 25%, and another with a weight of 20%, for a total of 45%.

Evaluation task	%	Approximate date	Link to competency(ies) and element(s)	Select if part of the final evaluation!
In Class Exercises	20		1-4	☐
Assignment 1	10	Class 4	1-2	☒
Test 1	20	Class 5	1-2	☒
Assignment 2	10	Class 9	1-4	☒
Test 2	25	Class 10	1-4	☒
Project	15	Class 12	1-4	☒
				☐
				☐
				☐
				☐

E. Course Content and Schedule

Course Content

Intro to Programming and Java
- Java language specification, API, JDK, and IDE
- The basic syntax of a Java program
- Create, compile, and run a Java program, Java IDE
- Programming conventions and documentation standards
- The differences between syntax errors, runtime errors, and logic errors

Language Basics
- variables, constants, methods, and classes
- assignment statements and assignment expressions
- primitive data types: byte, short, int, long, float, and double
- operators +, -, *, /, and %, exponent operations using Math.pow(a, b)
- integer literals, floating-point literals, and literals in scientific notation
- system time using System.currentTimeMillis()
- pre increment and post increment, pre decrement and post decrement
- cast the value of one type to another type

Equality, Relational, Conditional Operators, and If Statements
- boolean variables and expressions using relational operators
- if statements, if-else statements, nested if statements
- random numbers using the Math.random() method
- logical operators (&&, ||, and !), operator precedence
- switch statements
- debugging techniques

Mathematical Functions, Characters, and Strings
- the Math class
- the char type
- encode character sets ASCII and Unicode
- escape sequences
- conversion between types
- the Character class
- the Strings class, length() method, charAt(i) method
- concatenate strings, compare strings, substrings
- indexOf method
- System.out.printf method

Loops
- while loop, sentinel-controlled loop, do-while loop
- for loop, nested loops, break/continue

Java Methods, Parameters, Scope
- methods with formal parameters, methods with actual parameters (i.e., arguments)
- methods with a return value, methods without a return value
- pass arguments by value, scope of variables
- reusable code, method overloading,
- method abstraction in software development
- methods using stepwise refinement

Arrays

declare array reference variables and create arrays, obtain array size using arrayRefVar.length
default values in an array, access array elements using indexes
declare, create, and initialize an array using an array initializer
common array operations (displaying arrays, summing all elements, finding the minimum and maximum elements, random shuffling, and shifting elements)
the foreach loop
develop and invoke methods with array arguments and return values
define a method with a variable-length argument list
search elements using the linear or binary search algorithm.
sort an array using the selection sort approach
the java.util.Arrays class
pass arguments to the main method from the command line

Source Control

using git for tracking and managing source code
local repository:
creating a repository, checking status, adding files, committing files, removing staged files, removing committed files, checking logs
remote repository: init, clone, push, pull

Schedule

Date or class	Topic(s)	Additional info	F2F	Online
1	Intro to Programming and Java	Java language specification, basic syntax, create, compile, and run a java program, Programming conventions, Types of errors, Java IDE	☐	☐
2	Language Basics	variables, constants, methods, and classes assignment, data types, operators, literals, system time, increment and decrement, cast	☐	☐
3	Equality, Relational, Conditional Operators, and If Statements	boolean variables if, if-else, nested if Math.random() method logical operators, operator precedence switch statements debugging techniques	☐	☐
4	Mathematical Functions, Characters, and Strings Assignment 1 Due	Math class, char type, encode character sets, escape sequences,	☐	☐

		conversion between types, Character class, Strings class, compare strings, Substrings, indexOf, System.out.printf		
5	Test 1		☐	☐
6	Loops	while loop, sentinel-controlled loop, do-while loop, for loop, nested loops, break/continue	☐	☐
7	Java Methods, Parameters, Scope	Parameters, arguments, return value, pass by value, reusable code, method overloading, scope of variables, method abstraction, stepwise refinement	☐	☐
8	Arrays	declare and create arrays, arrayRefVar.length, default values, indexes, array operations, foreach loop, array arguments and return values, variable-length argument list, linear or binary search, selection sort, java.util.Arrays, pass arguments from the command line	☐	☐
9	Source Control Assignment 2 Due Review	using git for tracking and managing source code	☐	☐
10	Test 2		☐	☐
11	Project		☐	☐
12	Project Due		☐	☐

F. Required Textbooks / Materials / Costs

Title / Item	Cost $

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.

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/Project: Case problems, concepts reviews, and skills practice, will help support and reinforce material in the course. These will be structured to be as realistic as possible given the time available.
- Tests
- Team Project: The project focuses on methodologies and tools seen in this course. This project is structured to be small, but somewhat realistic given the time available in the course.
- 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

J. College Policies

Please refer to the following document which summarizes some of the key policies in place at the College. See the specific policies for more information.

Summary of College Policies and Guidelines

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)

Topic	Policy or Guideline (click link)	Article (if applicable)
Student Rights and Responsibilities	Policy 7: Institutional Policy on the Evaluation of Student Achievement (IPESA)	See articles 3.2 and 3.3.
Changes to Course Evaluation Plan in the Course Outline		See article 3.1 and 5.3.
Religious Holidays		See articles 3.2 and 4.1.
Cheating and plagiarism		See articles 9.1 and 9.2.
	Academic Integrity: Cheating and Plagiarism Procedure	
Student Code of Conduct	Policy 13: Policy on Student Conduct and Discipline Procedures (September 15, 2009)	