Data Structure and Algorithm Course Outline

IPD

A. General Information

Course name	**Data Structures and Algorithms**
Course number	420-PE3-AB
Start date	April 21, 2021
End date	May 5, 2021
Times	08:30am – 02:00pm (with 1/2 hour for lunch)
Classroom number	online
Ponderation *Ratio of lecture, practical and homework hours*	1-2-3 15 hours lecture – 30 hours lab – 45 hours homework
Hours	45
Credits	2.00
Competency statement(s) and code(s)	DC76: Apply data structures and algorithms to programming
Prerequisite (if any)	Java II
Semester	Winter 2021
Teacher's name	Gregory B. Prokopski, PhD
Teacher's contact info	MIO

B. Introduction

This course is part of the IPD program leading to the AEC. It should be taken in the second semester of the program.
Using a computer workstation connected to the Internet, the student will improve the programming skills learned previously by adding a whole new paradigm: Object-Oriented Programming. Using various technical and non-technical tools, such as the class and sequence diagrams, the student will improve his/her knowledge of Java and the JVM (Java Virtual Machine. The main focus of this course will be learning more advanced algorithms, classes, interfaces, inheritance, polymorphism, as well as design patterns frequently used in application development and API development.

C. Course Objectives

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

- Distinguish the steps to the realization of a software product
- Solve a problem in algorithmic form
- Develop a simple program using the basic elements of a structured language
- Develop a complex program using the advanced elements of a structured language

Competency

DC76	
OBJECTIVE	**STANDARD**
Statement of the Competency Apply data structures and algorithms to programming	**Achievement Context** • Based on situations representative of the workplace • Using a workstation and the appropriate software • Starting from a problem in a given situation • Using nomenclature and coding rules • Using appropriate technical reference manuals for the programming environment
Element	**Performance Criteria**
1. Analyze the problem	1.1 Correct specification of input data 1.2 Correct specification of output data 1.3 Correct specification of the nature of the procedure 1.4 Correct identification of the conditions for executing the algorithm 1.5 Proper identification of the algorithms to be created
2. Organize data in memory	2.1 Analysis of the context in which the data is to be used 2.2 Comparison of the features of the different data structures 2.3 Choice of appropriate data structures 2.4 Correct creation of arrays, linked lists, stacks, queues, lists, trees in a programming language
3. Model the classes	3.1 Proper identification of class attributes and methods 3.2 Proper application of encapsulation and inheritance principles 3.3 Proper graphic representation of the classes and their relationships 3.4 Compliance with nomenclature rules
4. Produce algorithms for the methods	4.1 Choice of a way to represent algorithms that is in accordance with company requirements 4.2 Definition of a logical sequence of operations 4.3 Identification of processing structures appropriate for each operation 4.4 Search for an effective algorithmic solution 4.5 Precise representation of the chosen algorithmic solution 4.6 Inclusion of all data necessary to interpret the algorithm 4.7 Appropriate verification of algorithm correctness 4.8 Accurate representation of algorithms
5. Translate the algorithm into the programming language	5.1 Effective use of environment editing features 5.2 Application of the syntax and semantic rules specific to the language used 5.3 Rigorous application of coding standards 5.4 Appropriate application of the principles of structured programming 5.5 Judicious exploitation of the language's possibilities 5.6 Recording of pertinent comments in accordance with business requirements
6. Program the classes	6.1 Appropriate choice of instructions, types of primitive data and data structures
7. Compile the program	7.1 Effective use of the environment's compilation features 7.2 Detection of compilation errors 7.3 Correction of compilation errors
8. Validate the program	8.1 Efficient use of the environment's execution and debugging features 8.2 Correct preparation of the test cases necessary for the verification of the functioning of the program

8.3	Accurate interpretation of the results
8.4	Appropriate debugging of the program according to the algorithm
8.5	Verification of the pertinence of the solution, given the initial situation
8.6	Identification of the errors and deficiencies of the algorithmic solution developed
8.7	Appropriate modification of the algorithmic solution

D. Evaluation Plan

			DC76							
Evaluation	%	Session/Date	1	2	3	4	5	6	7	8
Midterm	40	4 or 5	x	x	x	x	x	x	x	x
Final exam	60	9	x	x	x	x	x	x	x	x
Total	100%									

E. Recommended Textbooks / Materials

Title / Item Name	Cost
Head First Java, by Kathy Sierra, Bert Bates	43 $
Java How to Program, by Paul J. Deitel (Author), Harvey Deitel	115 $
Intro to Java Programming, by Y. Daniel Liang	189$

F. Bibliography (if applicable)

G. Teaching Methods

The course is a combination of theory and practical work. Students will be required to:

- Listen to lectures **and taking notes**
- Watch demonstrations
- Accomplish regular work in the laboratory
- Work in groups of 1 to 3 students for a project

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.

Lectures/Demonstrations: Important material from the text and outside sources will be covered in class. You should plan to take careful notes as not all material can be found in the texts or readings. Discussion is encouraged as is student-procured, outside material relevant to topics being covered.

- Assignments: Review Assignments, Case Problems, Concepts Reviews, Skills Reviews,

Independent Challenges and other projects and readings will be periodically assigned to help support and reinforce material in the course. These assignments may require the application of various software applications.

- Assignment submission: Assignments submission can be done with Lea or a Cloud Source Control tool, as detailed in class.
- Tests: The exams will be closed book/note and will test assigned readings and material discussed in class.
- Practical Test: The practical test will be on a lab computer only, with access to all online documentation, but no communication between students (be it electronic or verbal).
- Team Project (if any): 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
- JAC Portal: All material will be distributed on the JAC Portal. Class notes, instructional material, and student assignments will be posted on the class website. Students are encouraged to go to the website https://johnabbott.omnivox.ca/ in order to obtain file downloads and view other items of interest throughout the semester.

H. Departmental and Classroom Policies

Centre for Continuing Education Classroom Behaviour Policy

Class time is limited, and each student at John Abbott College is entitled to the very best educational experience in every course. You are expected to behave in a way that is civil and courteous to others. It is important that the atmosphere of each classroom or computer lab be as conducive to the learning process as possible. The following guidelines have been established in order to create and maintain such an atmosphere.

Inappropriate behaviour in the classroom includes the following:

- Using computer devices or other electronic devices unrelated to the course.
- Searching the internet or reading electronic materials unrelated to the course.
- Speaking while another person (teacher or student) has the floor (that is, he/she is addressing the class as a whole).
- Asking questions or making comments that are unrelated to the discussion at hand.
- Working on homework for other courses or other personal activities during class.
- Threatening, harassing, or offensive behaviour towards any person in the class, other students, teachers or College staff.
- Using derogatory language or referring directly or indirectly to someone else in the class in a rude manner or using offensive language.
- Misusing or abusing College computers, telephone systems or other equipment.
- Arriving late, leaving early, and leaving the room for any non-emergency without having teacher approval and the courtesy to make this known.
- Eating or drinking in the computer labs is discouraged.

A teacher is responsible for determining the appropriateness of student behaviour in the classroom. A teacher may remove a student who misbehaves in class for the duration of that period.

Centre for Continuing Education Attendance Policy

The College expects students to attend all class sessions. It is an essential requisite for their academic success and attainment of competencies. Excessive absences (over 20% of total course hours) may have consequences affecting the final course grade, including possible failure.

1. A student's attendance in class shall be excused if they provide written proof of a valid reason for missing a class, test or an evaluation due date.
2. Teachers are not required to re-teach course material missed by absent students. Students with excused absences cannot lose grades for missing a minor evaluation.
3. Teachers must provide alternate major evaluations if students miss a major evaluation due to an excused absence.
4. If a minor evaluation cannot be made up, the evaluation can be redistributed as long as all elements of the competency are assessed.
5. Absences of **less than 20% of total course hours** are addressed by the teacher and the student on a case-by-case basis.
6. Students who wish to observe religious holidays must inform their teachers, in writing, at the beginning of the semester so that alternative arrangements can be made between the teacher and student.
7. In cases of anticipated or planned absences for health or other reasons, students must request advance written approval for an excused absence from each teacher of their respective courses.

Centre for Continuing Education Late Submission of Work Policy

A teacher may deduct up to 10% per calendar day for late assignments that are submitted without a valid excuse.

I. College Policies

Policy No. 7 – IPESA, Institutional Policy on the Evaluation of Student Achievement (May 2017)

Cheating and Plagiarism (Article 9.1 & 9.2)

Cheating and plagiarism are unacceptable at John Abbott College. They represent infractions against academic integrity.

Students are expected to conduct themselves accordingly and must be responsible for all of their actions. The Academic Administration and teachers have the responsibility to:

• inform students of cheating and plagiarism as outlined below;

• teach all students what cheating and plagiarism are and inform them of the resulting consequences;

• determine whether cheating and/or plagiarism has occurred and take action according to the ACADEMIC PROCEDURE: Academic Integrity – Cheating & Plagiarism.

Cheating means any dishonest or deceptive practice relative to examinations, tests, quizzes, lab assignments, research papers or other forms of evaluation tasks. Cheating includes, but is not restricted to, making use of or being in possession of unauthorized material or devices and/or obtaining or providing unauthorized assistance in writing examinations, papers or any other evaluation task and submitting the same work in more than one course without the teacher's permission. It is incumbent upon the department through the teacher to ensure students are forewarned about unauthorized material, devices or practices that are not permitted.

Plagiarism is a form of cheating. It includes copying or paraphrasing (expressing the ideas of someone else in one's own words), of another person's work or the use of another person's work or ideas without acknowledgement of its source. Plagiarism can be from any source including books, magazines, electronic or photographic media or another student's paper or work.

Religious Holidays (IPESA Art 3.2.13 and 4.1.6)

Students who wish to miss class to observe a religious holiday, must inform the teacher in writing by the second day of class.

Student Rights & Responsibilities (IPESA Art 3.2.18)

It is the fundamental responsibility of each student to be a full and active participant in his or her education. Students have the responsibility to keep a copy of all assessed material returned to them and/or all digital work submitted to the teacher for at least four (4) weeks past the grade submission deadline of each individual course, in the event that they request a Final Grade Review (Refer to Article 8).

Changes to Course Evaluation Plan (Art.5.3)

Major changes (i.e. weighting, type and number of assessments) can be made to the course evaluation plan (on the course outline) due to exceptional circumstances. To do so, the teacher must ensure that any major changes to the evaluation plan made during the semester be forwarded (on paper or electronically) the AEC program coordinator for approval. All changes must have documented unanimous consent from the regularly attending students affected by the change(s) before submission. The approved major change will then be communicated to students on paper or electronically.

Session	Content	Link to Competency
1	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)	DC76: 1,2,3,4,5,6,7,8
2	lists, stacks, queues, priority queues (stack machine language example) HashMap, LinkedHashMap, TreeMap HashSet, LinkedHashSet, TreeSet	DC76: 1,2,3,4,5,6,7,8
3	singly linked list, doubly linked list bubble sort algorithm example unit testing	DC76: 1,2,3,4,5,6,7,8
4	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	DC76: 1,2,3,4,5,6,7,8
5	**Midway test**	DC76: 1,2,3,4,5,6,7,8
6	recursion (tail recursion, fractals, directory scanning) graphs and trees - their implementation and use	DC76: 1,2,3,4,5,6,7,8
7	Java programming patterns (not all mentioned may be covered): - factory pattern - singleton pattern - visitor pattern - iterator design pattern	DC76: 1,2,3,4,5,6,7,8
8	Java programming patterns (not all mentioned may be covered): - observer pattern - functional strategy pattern - service callback pattern	DC76: 1,2,3,4,5,6,7,8
9	**Final exam**	DC76: 1,2,3,4,5,6,7,8

Note: The above schedule is tentative. Not all items listed will be covered in the course. Teacher reserves the right to modify the order of items listed and amount of time spent on each item, as well as substitute one item for another as long as it fulfills the requirements of the competency.