Dawson College

# **Computer Science Technology Program**

COURSE OUTLINE

## **Software Development Project**

## **Java III**

**Number: 420-517-DW Ponderation: 3-3-3 Credits: 3**

**Prerequisites:**  Successful completion of all Year 2 420.A0 courses

**Co-requisites:** 420-518-DW Mobile Software Development: Android

420-524-DW Internet Application Development III: PHP

420-544-DW Data Communications and Networking

**Domain:** Programming, Data Structures and Algorithms

**Note:** You must pass this course as well as the three co-requisite courses to progress to the sixth semester of the Computer Science Technology Program.

### Semester: Fall 2020 Revision: 2020-07-25

### Description

In this course you will learn to design, code and test a desktop application. The application will employ a JavaFX graphical user interface. The data for the application will be stored in and retrieved from a relational database using the Java Database Connectivity library. All components of the application will be tested using the JUnit framework. You will learn the purpose of, the application of and how to code common data structures. You will also review the data structures provided in the Java Collections Framework. These data structures will be applied to the development of code that implements algorithms. You will learn to use a build system to manage your code a repository to store your code.

**STATEMENTS OF COMPETENCE**

* Use an object-oriented development approach (016T)
* Produce algorithms (016W)
* Develop user interfaces. (016X)
* Organize and use data (0170)
* Correct programs (0171)
* Make functional improvements to an application (0176)
* Assure the quality of an application (0177)
* Design and develop applications in a graphics environment (017C)

**Teacher: Office: Local: Section:**

Ken Fogel 3F.18 4799 01,02

Please use [kfogel@dawsoncollege.qc.ca](mailto:kfogel@dawsoncollege.qc.ca) for email contact.

**Course Objectives:**

Upon completion of this course the student will be able to:

1) Develop Java applications that use JavaFX framework (016X)

2) Connect a Java application to a database by utilizing Java Data Base Connectivity (JDBC) libraries (0170)

3) Explain the concepts for and apply practically in an application data structures such as stacks, lists, queues, linked lists and binary trees (016W)

4) Code stacks, lists, queues, and linked lists (016W)

5) Use the Java Collections Framework to employ data structures in an application (0170)

6) Issue SQL commands to a database and display the result set in an appropriate GUI component such as an Update Table (0170)

7) Manage application development using a build system

8) Test and validate your code using the unit testing framework (0171, 0177)

9) Write a desktop application from a given specification (016T, 017C, 0177, 016W)

**Course Methodology:**

The course consists of two 1½-hour lectures and one 3-hour lab session per week. Concepts and techniques will be presented in lectures and practical exercises that apply to the concepts and techniques will be carried out in lab sessions.

Due to the pandemic all sessions will be virtual and will consist of a combination of videos and live Zoom presentations. During labs each student is expected to have Zoom running. Questions may either be public or in a breakout room between you and the instructor.

**Required Text:**

Java By Comparison

Become a Java Craftsman in 70 examples

By Simon Harrer, Jorg Lenhard, Linus Dietz

Copyright 2018

Published by The Pragmatic Programmers, LLC

ISBN-13:978-1-68050-287-9

**Required:**

1. An account on GitLab.com (not GitHub or any other service).
2. Four Gmail accounts for testing of the email application.

**Software**

1. Java 14

<https://adoptopenjdk.net/?variant=openjdk14&jvmVariant=hotspot>

1. NetBeans 12

<https://netbeans.apache.org/>

1. MySQL 8

<https://dev.mysql.com/downloads/mysql/>

**Assessment of student performance:**

**Exam 1** Duration: 1 1/2 hours 20% (Week 7)

**Exam 2 (Coding)** Duration: 3 hours 30% (Exam period)

**Major Project** 35%

**Minor Project** 15%

**Course Topics & Labs**

|  |  |  |
| --- | --- | --- |
| **Lectures** | **Topic** | **Lab Work** |
| **1** | **Presentation of the course project** | **Java Application for GMail** |
| **1** | **Software Build System** |  |
|  | * Learn to use Maven to manage the building of Java application | Create projects that use Maven |
|  | <https://maven.apache.org/> |  |
| **2** | **Mail** |  |
|  | * Review the principles of SMTP, POP, and IMAP systems. |  |
|  | * Learn to use the JODD Email API to connects to SMTP and IMAP mail systems to send and retrieve emails | Implement the email portion of the project |
| **2** | **JUnit Testing** |  |
|  | * **Examine how the JUnit Framework supports unit testing** |  |
|  | * **Learn to develop test cases** | **Develop a test plan and implement it** |
|  | <https://junit.org/junit5/docs/current/user-guide/index.pdf>  <https://cguntur.me/2019/07/07/using-junit5-part-1/>  <https://cguntur.me/2019/07/13/using-junit5-part-2/>  <https://cguntur.me/2019/07/20/using-junit5-part-3/>  <https://cguntur.me/2019/07/27/using-junit5-part-4/> |  |
| **2** | **JDBC** |  |
|  | * Setup and configure the MySQL RDBMS |  |
|  | * Examine the principles of interacting with a DBMS from within a program. |  |
|  | * Examine the JDBC interface and classes that support database access. |  |
|  | * Implement the tasks necessary to:   + Connect to a database   + Issue an SQL statement   + Process the result | Implement a database for storing tweets |
|  | <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html> |  |
| **6** | **JavaFX** |  |
|  | * Examine the architecture of JavaFX   + Scene Graph   + Java Public APIs for JavaFX Features   + Graphics System   + Media and Images   + Web Component   + CSS   + UI Controls   + Layout | Create form-based user interfaces |
|  | * Working with the Scene Builder tool to use FXML to code user interfaces and implement MVC   + View created with FXML   + Model implemented as a Properties Bean   + Controller implemented in Java objects that defines the behavior of the GUI | Create multi-panel interface with table, trees, forms. Menu and toolbar. |
|  | * Apply data validation to user input. | . |
|  | <https://openjfx.io/>  <https://fxdocs.github.io/docs/book.pdf> <https://code.makery.ch/library/javafx-tutorial/>  <https://docs.oracle.com/javase/8/javafx/JFXST.pdf> |  |
| **2** | **Internationalization and Properties** |  |
|  | * Discuss how software is designed to work in multiple languages and locales. |  |
|  | * Examine the approach Java takes to support software that can dynamically change in different locales. | Implement internationalization in a JavaFX application |
|  | * Examine how programs can maintain configuration information |  |
|  | * Examine the Java Properties class to store and retrieve configuration data | Implement a Properties file. |
|  | <https://docs.oracle.com/javase/tutorial/i18n/>  <https://docs.oracle.com/javase/tutorial/essential/environment/properties.html> |  |
| **2** | **Concurrency and threads** |  |
|  | <https://www.baeldung.com/java-concurrency> |  |
|  | **The Java Collections library** |  |
| 3 | **Sequential Data Structures** |  |
|  | * Examine the sequential structures   + Without hash codes   + With hash codes | Implement a GUI calculator that uses an equation parser with stacks and queues |
|  | * Examine the hash structure |  |
| **2** | **Lambdas & Streams** |  |
|  | * **Examine and apply the concepts of Functional programming** |  |
| **3** | **Non-sequential data structures** |  |
|  | * Examine the map structures   + TreeMap   + HashMap |  |
|  | * Review the Big-O performance characteristics of data structures |  |
|  | * Compare the performance of the various collections classes to determine the best choice for an algorithm. |  |

**Readings:**

Java by Comparison is just 164 pages broken into 9 chapters of about 20 or fewer pages each. You are expected to have finished reading the book before the mid term exam if not sooner. There will be several questions on the material in the book on the mid term.

As we cover topics in the course you will find references to readings in this outline and I will provide you with additional recommended readings from the web. Doing this reading is critical to your success in this course.

**Lab Activities:**

**Major Project**

During the term you will work individually on a project that will demonstrate the achievement of each of the objectives in this course. The project will be broken into four phases.

For Fall 2020 the project will be an Email Client program. The four phases of this project will be:

1) Design, implement, and unit test the email access classes and data classes.

Due on **September 20, 2020**

2) Design, implement, and unit test the database access objects and data objects Due on **October 5, 2020**

3) Design and implement a graphical user interface

Due on **October 24, 2020**

4) Integrate the three subsystems into the finished application.

Due on **November 15, 2020**

Phases 1 and 2 must include unit tests. These tests will remain a part of the project till the end.

The project will be completed by **November 15, 2020** or sooner. ***There will be no extension.*** Each phase is worth 8.75 marks of the total project mark of 35 marks.

If phases 1, 2 or 3 do not meet the requirements of that phase and you receive a grade of less than 6 out of 8.75 then it may be resubmitted once as part of the next phase and can receive a maximum of 7 marks.

Failure to achieve at least 21/35 for the project will result in a failure in the course with a final grade of 50% (ISEP June 6, 2019, section IV, subsection 6, Grading Scheme, fourth bullet point)

**Minor Project**

Implement a GUI four function calculator that implements an equation parser using data structures from the Collections framework. Project is due **December 22, 2020**

**Program rules and regulations:**

***For 420 career program courses, students must receive at least 60% on the assignment component and at least 60% on the examination component in order to obtain a passing grade for the course. Failure to do so will result in a maximum grade of 50 (the lower of 50 or the grade earned). For the purpose of this rule, quizzes are considered exams.***

***A minimum grade of 60% is required to pass the course.***

Assignments are due at the due date specified by the instructor. Assignments received after this time are late. Assignments will not be accepted after due date set by instructor. In exceptional circumstances, such as illness, etc. labs submitted late will be graded provided that the student has received an extension from the instructor in advance.

Each student must submit their own individual answers to assignments, and shared solutions are not acceptable. Students must protect their work from being easily copied by others by introducing uniqueness into their work in the choice of images used, the display styles used in the presentation of documents, the originality of the written descriptive passages, and by placing digital signatures on their original images.

Students are reminded that they are responsible for all material presented in the lecture and during the lab and are expected to obtain material they have missed.

Students must follow exactly the instructions in the lab exercises about how and where to store all computer files in order for their work to be accepted and properly evaluated.

*“This course outline has been prepared in conformity with the* ***Institutional Student Evaluation Policy (I.S.E.P.)****. The “policy is designed to promote equitable and effective evaluation of student learning” and is* ***therefore a crucial policy for you to read and understand****. The policy describes the rights and obligations of students, faculty, departments, programs, and the College administration with regard to evaluation in all your courses.* ***I.S.E.P.*** *is published in the College Calendar and reference copies are available in the Library and in the offices of program chairs and the Dean of Social Science & Business Technologies”.*

**DAWSON COLLEGE**

**COMPUTER SCIENCE DEPARTMENT**

**POLICIES**

**1. ACADEMIC INTEGRITY**

The Computer Science Department adheres to the Dawson College Academic Conduct policy. Students have an obligation to inform themselves of all aspects of this policy. According to ISEP, the teacher is required to report to the Sector Dean all cases of cheating and plagiarism affecting a student’s grade. (ISEP Section IV-C) Students may appeal any decision of the teacher regarding cheating and plagiarism according to the methods established by the College. The penalty for cheating or plagiarism may range from a zero on the work to a failure in the course.

**2. ATTENDANCE AND LATENESS**

Students are responsible for all material covered in classes and labs, whether or not they are present. Students have an obligation to arrive on time and to remain for the duration of scheduled classes, labs and activities. Students who disregard this obligation may be asked to leave the class, lab or activity. Students should refer to the Institutional Student Evaluation Policy (ISEP Section III-C) regarding attendance.

**3. RELIGIOUS HOLIDAY OBSERVANCE**

Class period(s) may be cancelled in order that the teacher can observe religious holidays. Any material missed as a result, will be made up during labs, class periods and alternate tutorial sessions.

Students, who intend to observe religious holidays, must inform their teachers in writing as prescribed in the ISEP Policy on Religious Observances (ISEP Section III-D) Students absent from classes because of observance of religious holidays will not be penalized. It must be emphasized, however, that this College policy should not be interpreted to mean that a student can receive credit for work not performed. It is the student’s responsibility to fulfill the requirements of the alternative arrangement.

**4. SOFTWARE**

The Computer Science Department forbids the use of the computer labs it uses to make any copies of any software without the explicit authorization of the Department. Apart from any legal action that might arise from such unauthorized copying, the Computer Science Department reserves the right to discipline any student involved in such activity.

Students who borrow software from the College and fail to return it will be placed on the defaulter list and be subject to the appropriate penalties.

Viruses are programs that attach themselves to a computer system without the permission of those to whom the system belongs. They are deliberately written to be, at worst, harmful (e.g. destroying the contents of disks) and, at best, bothersome (e.g. disturbing the image shown on the screen). They can cause serious losses of time and effort for students, staff and faculty. Any student involved in the deliberate spreading of viruses is subject to the most severe penalties prescribed by College regulations, apart from any legal action that might arise from such acts.

**5. LITERACY**

The Computer Science Department recognizes that literacy in all its forms (read, written, spoken) is essential to our students in their careers.

Teachers may choose to incorporate a literacy component into the marking scheme for any piece of work. Teachers may use their discretion to insist that any piece of work submitted for credit is revised by the student if it is unsatisfactory with regard to literacy.

Teachers will inform all students in their courses of this policy at the beginning of each semester either by including it in the course description or otherwise.

**6. CELL PHONES**

The use of cell phones in all Computer Science lectures, labs and exams is prohibited.

**7. PORTABLE COMPUTERS**

The use of portable computers in all Computer Science lectures for purposes other than note-taking is prohibited.

**8. STUDENT CONDUCT**

Everyone has the right to a safe and non-violent environment. Students are obliged to conduct themselves as stated in the Student Code of Conduct and in the ISEP section on the roles and responsibilities of students. (ISEP section II-D)

**9. INTENSIVE COURSE CONFLICTS**

If a student is attending an intensive course, the student must inform the teacher, within the first two weeks of class, of the specific dates of any anticipated absences.

**10. ISEP**

The Institutional Student Evaluation Policy (ISEP) is designed to promote equitable and effective evaluation of student learning and is therefore a crucial policy to read and understand. The policy describes the rights and obligations of students, faculty, departments, programs, and the College administration with regard to evaluation in all your courses, including grade reviews and resolution of academic grievance. The ISEP is available on the Dawson website.