



Computer Science Technology 420.B0

420-6A6-AB COURSE OUTLINE

GENERAL INFORMATION

Program	Computer Science Technology 420.B0	
Course Title	Application Development III	
Course Number	420-6A6-AB	
Timetable and Classroom	Section 1: [Mon 8:30 am – 11:00 am & 03:30 pm – 04:00 pm] [Fri 8:30 am – 10:30 am & 11:30 am – 12:30 pm] Section 2: [Mon 8:30 am – 11:00 am & 03:30 pm – 04:00 pm] [Fri 8:30 am – 10:30 am & 11:30 am – 12:30 pm]	
Ponderation	2 hours lecture + 4 hours laboratory + 4 hours homework	
Number of Credits	3.33	
Competency and Code	Develop native applications with a database 00SS.	
Prerequisites	Application Development II (420-5A6-AB)	
Semester	Winter 2024	
Teachers		
Younna Badawy (Penfield 235) Contacts: <ul style="list-style-type: none">• youmna.badawy@johnabbott.qc.ca• MIO Office Hours: Posted on office door. Appointments can be made outside of the posted hours when the need arises.	Micheal Haaf (Penfield 311) Contacts: <ul style="list-style-type: none">• michael.haaf@johnabbott.qc.ca• MIO Office Hours: Posted on office door and LEA. Appointments can be made outside of the posted hours when the need arises.	

INTRODUCTION

This course covers the process of software development, consisting of the requirement gathering proposal, design, implementation, documentation, testing and deployment of a multiplatform software application. Students will learn the skills necessary to create applications that can run on multiple platforms: Windows, Android, MacOS, iOS, Chrome OS and/or Linux. Students will be exposed to one of the cross-platform app development (CPAD) such as: .NET MAUI, Flutter, React Native or Codename One.

COURSE OBJECTIVES

Competency 00SS. Develop native applications with a database.

Achievement Context

- Fully participate in the design and development of 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 competency	Performance criteria
00SS.5 Program the application logic.	<ul style="list-style-type: none">• 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 exchange services• Proper application of internationalization techniques• Precise application of secure programming techniques
00SS.6 Control the quality of the application.	<ul style="list-style-type: none">• 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
00SS.7 Participate in the deployment of the application.	<ul style="list-style-type: none">• Appropriate preparation of the application in view of its deployment or installation• Proper deployment or installation of the application
00SS.8 Produce the documentation	<ul style="list-style-type: none">• Proper identification of the information to be written up• Clear record of the work carried out

EVALUATION PLAN

Objective:	Weight	00SS
Assignments (4-6)	30%	X
Tests (1-2: Week 6 and/or Week 13) or Quizzes (4-6)	20%	X
Project: Milestones (3-5)	50%	X

(Note: Weeks indicated for Tests and Project are tentative)
Final evaluation consists of the project and the assignments.

COURSE CONTENT

General Learning Objective	Specific Learning Objective
1. Cross Platform App Development (CPAD)	1.1. Understand difference between native and cross platform development. 1.2. Examine challenges in CPAD. 1.3. Explore advantages of CPAD. 1.4. Research and compare different CPAD frameworks.
2. Setup and use of development environment.	2.1. Setup development IDE of chosen CPAD framework. 2.2. Setup & configure emulators for different target platforms to run and test apps.
3. Architect solutions for CPAD	3.1. Creation of shared code projects. 3.2. Create and use class libraries. 3.3. Plan for design-pattern (for example MVC / MVVM).
4. Create a shared UI.	4.1. Identify basic UI components. 4.2. Identify different layouts, views, and controls. 4.3. Customize components using properties. 4.4. Construct a responsive UI using layout managers. 4.5. Interact programmatically with components. 4.6. Design for accessibility issues.
5. Add functionality and interactivity	5.1. Understand navigation process. 5.2. Configure app settings and start up behavior. 5.3. Request appropriate app permissions from the user. 5.4. Access different device resources: files, camera, sensors, etc.
6. Access data in app	6.1. Examine different data access classes. 6.2. Consume web server services with CRUD capabilities (Create-Read-Update-Delete). 6.3. Use databases: 6.3.1. Cloud databases: such as Firebase. 6.3.2. SQLite/CoreData. 6.4. Access local files.
7. Apply authentication and authorization mechanisms	7.1. Differentiate between authentication and authorization. 7.2. Understand user authentication process. 7.3. Identify providers, relying parties, tokens, and claims. 7.4. Examine common authentication and authorization protocols: e.g., OAuth and/or OpenID.
8. Integrate with IOT devices (Optional)	8.1. Communicate with IOT devices to read or write data.

	8.2. Control IOT devices from app.
9. Use native platform integrations (Optional)	9.1. Apply custom UI modifications on specific platforms (OSs). 9.2. Apply custom UI modifications on specific idioms: desktop, tablet, or phone. 9.3. Use of native controls and renders.
10. App Testing	10.1. Write and execute UI tests using automated UI acceptance testing framework. 10.2. Write and execute unit tests to validate functionality of app on different platforms.
11. App Documentation and Deployment	11.1. Produce code documentation. 11.2. Learn how to generate .apk files for Android, .ipa files for iOS and other target platforms. 11.3. Explain how to deploy apps on different application stores: Google Play, App Store, Microsoft Store, etc.

TENTATIVE SCHEDULE

Week 1	Cross Platform App Development (CPAD) Frameworks
Week 2	Development Environment / Shared Code / Library Classes / UI Components
Week 3	UI Components (Buttons, controls, sliders, collections, etc.)
Week 4	Design Patterns: Publisher Subscriber Pattern - Data Binding: INotifyPropertyChanged
Week 5	Design Patterns: Singletons – Transients - Dependency Injection
Week 6	Design Patterns: MVVM Test 1
Study Break	
Week 7	Advanced Programming Concepts: Multithreading and Asynchronous Programming
Week 8	App settings / Navigation / Permissions / Resources / Native platform integrations and controls. / Android Apps and iOS Apps
Week 9	App Data: Local files (JSON)/ Databases
Week 10	App Data: Local files (JSON)/ Databases
Week 11	Access data in app: Web Services
Week 12	App Testing / Test plans / Documentation
Week 13	Data Representation using Graphs
Week 14	Authentication and authorization mechanisms
Week 15	Project
Week 16	Project
Week 17	Project Presentations
Week 15	Deployment / Documentation / Project Test 2
Week 16	Project and presentations

SUGGESTED TEXT

TBD

COURSE COSTS

TBD

TEACHING METHODS

This course consists of 90 hours of scheduled lectures and lab work. In addition, each student will be required to do 60 hours of personal study that includes research, personally booked computer time and work at home. Léa, the course management system within Omnivox, will be used in this course.

DEPARTMENTAL LATE POLICY

All assignments and projects are expected to be submitted by the required due date. A late penalty of up to 10% per day might apply to assignments submitted late up to a maximum number of allowed late days. The exact percentage and the maximum will be indicated on the instructions specific to that assignment. Any work submitted after the maximum number of allowed late days has been reached will not be graded.

DEPARTMENTAL ATTENDANCE POLICY

See Article 4 of the IPESA, [Policy-7-IPESA.pdf](#)

Active learning of competencies requires hands on learning with interactive classroom work, which requires no more than 20% absences.

Attendance is compulsory for the following program activities:

- Stage participation. Competency 016V, 016Y.
- Stage evaluation meetings. Competency 016Z.
- In-class group or project integration.

Unexcused absences of more than 20% of any of these activities will result in a grade of 0 for the assessment related to that activity.

COLLEGE POLICIES - IPESA, Institutional Policy on the Evaluation of Student Achievement: [Policy-7-IPESA.pdf](#)

- **Changes to Evaluation Plan in Course Outline** (Article 5.3)
Changes require documented unanimous consent from regularly attending students and approval by the department and the program dean
- **Religious Holidays** (Articles 3.2.13 and 4.1.6)
Students who wish to miss classes in order to observe religious holidays must inform their teacher of their intent in writing within **the first two weeks of the semester**

Student Rights and Responsibilities: (Articles 3.2.18 and 3.3.6)

- It is the responsibility of students to keep all assessed material returned to them and/or all digital work submitted to the teacher in the event of a grade review. (The deadline for a Grade Review is 4 weeks after the start of the next regular semester.)
- Students have the right to receive graded evaluations, for regular day division courses, within two weeks after the due date or exam/test date, except in extenuating circumstances. A maximum of three (3) weeks may apply in certain circumstances (e.g. major essays) if approved by the department and stated on the course outline. For evaluations at the end of the semester/course, the results must be given to the student by the grade submission deadline (see current Academic Calendar). For intensive courses (i.e. intersession, abridged courses) and AEC courses, timely feedback must be adjusted accordingly;

- **Academic Procedure: Academic Integrity, Cheating and Plagiarism** (Article 9.1 and 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.

College Definition of Cheating:

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.

College Definition of Plagiarism:

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.

For PowerPoint on cheating and plagiarism refer to the JAC Portal: My JAC Communities / Academic Council / Curriculum Validation Committee (CVC) / Course Outlines – Reference Documents / Academic Integrity.

For link to interactive tutorial on how to cite sources correctly: <http://citeit.ccdmd.qc.ca>