

Computer Science Technology 420.B0

420-5A6-AB

COURSE OUTLINE

## GENERAL INFORMATION

|  |  |
| --- | --- |
| Program | Computer Science Technology 420.B0 |
| Course Title | **Application Development II** |
| Course Number | **420-5A6-AB** |
| Timetable and Classroom | Section 1: Section 2: |
| Ponderation | 2 hours lecture + 4 hours laboratory + 3 hours homework |
| Number of Credits | 3.00 |
| Competency and Code | Develop native applications without a database 00SR. |
| Prerequisites | Programming IV (420-4P6-AB) and Application Development I (420-4A8-AB) |
|  | This course is a prerequisite for Application Development III (420-6A6-AB), Stage I (420-6S9-AB) and Stage II (420-6SC-AB) |
| Semester | **Fall 2022** |
|  |  |
| Teacher |  |
| Name | Gabriel Larco |
| Office | Penfield 235 |
| Phone | (514) 457-6610, local 5063 |
| Internet | gabriel.larco@johnabbott.qc.ca |
| Office Hours | Posted on office door. 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 proposal, design, implementation, documentation, and deployment of a software project. Students will learn the skills necessary to create a mobile application using the Android framework in Java.

## COURSE OBJECTIVES

**Competency 00SR. Develop native applications without a database.**

### Achievement Context

* 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, sounds and videos
* Using issue tracking and version control procedures

|  |  |
| --- | --- |
| **Elements of competency** | **Performance criteria** |
| 00SR.1 Analyze the application development project. | * Accurate analysis of design documents * Proper identification of the tasks to be carried out. |
| 00SR.2 Prepare the computer development environment. | * Proper installation of software and libraries on the development platform * Proper configuration of the target platform * Proper configuration of the version control system * Proper importing of the source code |
| 00SR.3 Generate or program the graphical interface. | * Appropriate choice and use of graphic elements for display and input * Proper integration of images * Adaptation of the interface based on the display format and resolution |
| 00SR.4 Program the application logic. | * Proper programming of interactions between the graphical user interface and the user * Proper programming of communications between the peripheral devices and the software functions of the target platform * Effective use of execution threads * Proper integration of sounds and videos * Proper application of internationalization techniques * Precise application of secure coding techniques |
| 00SR.5 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 design documents |
| 00SR.6 Participate in the deployment of the application. | * Appropriate preparation of the application in view of its deployment or installation * Proper deployment or installation of the application |
| 00SR.7 Produce the documentation | * Proper identification of the information to be written up * Clear record of the work carried out |

**EVALUATION PLAN**

|  |  |
| --- | --- |
| **Objective:** | **00SR** |
| Assignments (3-4) 30% | X |
| Quizzes (4-6) 15% | X |
| Project (4-5 milestones) 55% | 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 Setup and use the Android development environment. | 1.1 Setup up the IDE. |
| 1.2 Create and configure Android simulator(s) appropriate for the app. |
| 1.3 Monitor the simulator using the Android Debug Bridge, log viewer and profiler. |
| 2 Create simple Android UI. | 2.1 Identify the basic UI components (views). |
| 2.2 Modify components using properties. |
| 2.3 Construct a responsive UI using layout managers. |
| 2.4 Inflate layouts. |
| 2.5 Extract view references from the component tree. |
| 2.6 Interact programmatically with components. |
| 3 Develop an app that includes activities, fragments, and containers. | 3.1 Appropriately group an app into activities and fragments. |
| 3.2 Understand activity lifecycle and associated events. |
| 3.3 Specify the launching activity of an app. |
| 3.4 Embed multiple fragments into an activity. |
| 3.5 Use Intents to implement app navigation. |
| 3.6 Reuse fragments across activities. |
| 3.7 Incorporate dialog fragments in an app. |
| 3.8 Display a collection of data in a container UI component, ex: RecyclerView. |
| 3.9 Design and implement an interactive data adapter. |
| 3.10 Notify the adapter of data set changes. |
| 4 Design an application using event-driven programming. | 4.1 Add event handlers to UI components. |
| 4.2 Add events to fragments/dialogs using the callback pattern. |
| 5 Include custom UI components in an app. | 5.1 Design a custom component by extending an existing component. |
| 5.2 Draw the component using the drawing classes (Paint, Canvas, etc.). |
| 5.3 Add events to custom UI components using the callback pattern. |
| 6 Improve on basic app development. | 6.1 Request appropriate app permissions from the user. |
| 6.2 Use intents to access Android services (ex: phone). |
| 6.3 Read and write data to device's filesystem and/or the local database. |
| 6.4 Store app preferences. |
| 6.5 Incorporate custom features from elements of the Android framework. Ex: notifications, sensors, GPS, animation, etc. |
| 7 Perform asynchronous tasks in an app. | 7.1 Understand simple thread concepts. |
| 7.2 Implement background tasks using threads. |
| 7.3 Incorporate asynchronous tasks in the application design. |
| 8 Use build automation tools. | 8.1 Identify the tasks of the build automation tool. |
| 8.2 Add software dependencies. |
| 8.3 Include 3rd party libraries into an application. |
| 8.4 Create custom build tasks. |
| 9 Develop a complete Android app. | 9.1 Propose an idea for an app. |
| 9.2 Develop and evaluate an app prototype. |
| 9.3 Design and implement an app from the proposed idea. |
| 9.4 Use version control in development, including branches to develop new features and fix bugs. |
| 9.5 Produce code documentation. |
| 9.6 Understand the process for deploying an app. |

**TENTATIVE SCHEDULE**

|  |  |
| --- | --- |
| Week 1 | Android Development Environment |
| Week 2 | Android UI / Activities / Events |
| Week 3 | Events / App / Fragments |
| Week 4 | Lists |
| Week 5 | Lists/ Local Storage |
| Week 6 | Custom UI / Custom Events |
| Week 7 | Custom Events / Intents |
| Week 8 | Intents |
| Week 9 | Asynchronous tasks |
| Week 10 | Asynchronous tasks |
| Week 11 | Development |
| Week 12 | Project |
| Week 13 | Project / Build tools |
| Week 14 | Deployment / Documentation / Project |
| Week 15 | Project |

## 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 45 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](http://departments.johnabbott.qc.ca/wp-content/uploads/2017/08/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 PROVISOS – Fall 2022

* Attendance: Due to the ongoing pandemic health issues, attendance policies may need to be adjusted by your teacher. The normal attendance expectations are outlined above, and your teacher will inform you of any modifications as needed. Please note that attendance continues to be extremely important for your learning, but your teacher may need to define it in different terms based on the way your course is delivered during the semester.
* Please note that course outlines may be modified if health authorities change the access allowed on-site. This includes the possibility of changing between in-person and online formats.

**COLLEGE POLICIES -** IPESA, Institutional Policy on the Evaluation of Student Achievement: [Policy-7-IPESA.pdf](http://departments.johnabbott.qc.ca/wp-content/uploads/2017/08/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**

**Course Outlines – Provisos**

* **Attendance:** Due to the COVID-19 health crisis, attendance policies may need to be adjusted by your teacher. The normal attendance expectations are outlined below and your teacher will inform you of any modifications as needed. Please note that attendance continues to be extremely important for your learning, but your teacher may need to define it in different terms based on the way your course is delivered during the semester.
* Please note that course outlines may be modified if health authorities change the access allowed on-site. This includes the possibility of changing to an entirely on-site or online format.
* In addition to LEA, Teams and Moodle, additional software may be used for the submission of essays or projects or for testing. Further details will be provided if applicable.
* Classes on Teams may be recorded by your teacher and subsequently posted on Teams to help for study purposes only. If you do not wish to be part of the recording, please let your teacher know that you wish to not make use of your camera, microphone or chat during recorded segments. Any material produced as part of this course, including, but not limited to, any pre-recorded or live video is protected by copyright, intellectual property rights and image rights, regardless of the medium used. It is strictly forbidden to copy, redistribute, reproduce, republish, store in any way, retransmit or modify this material. Any contravention of these conditions of use may be subject to sanction(s) by John Abbott College.