

CÉGEP DE SAINT-HYACINTHE

3000, rue Boullé, Saint-Hyacinthe

Québec J2S 1H9

Téléphone : (450) 773-6800

Montréal : (514) 875-4445

Plan de cours

Trimestre Automne 2016

Développement d’applications en mobilité

420-GEP-HY

(3-3-3)

Nom de l’enseignant : Stéphane Denis et Sébastien Pouliot

Bureau : A-1402D

Poste téléphonique : SD : 2089, SP : 2490

Adresse électronique : par MIO

### OBJECTIF GÉNÉRAL

Au terme de ce cours l’étudiant sera en mesure de produire des applications efficaces et bien adaptées aux particularités des plateformes mobiles.

### LIENS AVEC LES AUTRES COURS ET

### PRÉALABLES OU CO-REQUIS

|  |  |
| --- | --- |
| 1ère session | 420-CAC-HY Algorithme et programmation orientée objet |
| 2e | 420-GEA-HY Programmation orientée-objet |
| 3e | - |
| 4e | 420-GEK-HY Architecture des applications |
| 5e | 420-GEP-HY Développement d’applications distribuées |
| 6e | (Stages) |

### DESCRIPTION SOMMAIRE DU COURS

Dans un premier temps, ce cours explore les caractéristiques techniques et les principaux usages des applications en mobilité.

Dans un second temps, une architecture et une approche de développement est sélectionnée et mise en pratique pour aborder les autres aspects du développement d’applications en mobilité.

La plateforme Android versions 4.4 à 7 et l’IDE **Android Studio 2.1** (IntelliJ) ont été retenus pour cette session. Les étudiants pourront utiliser leurs propres appareils Android ou en emprunter un au département pour toute la session.

### COMPÉTENCE ET ÉLÉMENTS DE COMPÉTENCES DU PROGRAMME

|  |  |
| --- | --- |
| **17D** | **Concevoir et développer une application hypermédia dans des réseaux internes et mondiaux**   1. Établir les fonctionnalités de l’application. 2. Établir le cadre technologique. 3. Préparer le travail de développement de l’application. 4. Produire le prototype de présentation. 5. Produire le prototype de communication. 6. Développer l’application. 7. Produire la documentation relative à l’application. |

### OBJECTIFS SPÉCIFIQUES

* Éventail des différents dispositifs mobiles et leurs systèmes d’exploitation :
  + Principaux systèmes d’exploitation en mobilité ;
  + Variabilité des formats (Tablette, téléphone, montre, voiture, …)
  + Variabilité des ressources systèmes ;
  + Variabilité des capacités et de la fiabilité de l’accès réseau ;
  + Variabilité des interfaces (Culture d’interface graphique, taille d’écran, vocal, tactile, ….
* Architecture et approches de développement applicatif :
  + Natif (C++, Objective-C, …) ;
  + Sandbox (Java, .Net, …) ;
  + Multiplateforme par navigateur (HTML5, CSS, JavaScript, …) ;
  + Multiplateforme par compilation native de code partagé (Xamarin).
* Cycle de développement des applications mobiles ;
* Design d’interface, ergonomie et expérience utilisateur flexibles ;
* Sécurité en mobilité ;
* Autonomie hors-ligne à l’aide de bases de données ultralégères ;
* Synchronisation des données avec des services web.

| **Savoirs essentiels** |
| --- |
| * Produire une analyse fonctionnelle * Produire un document de conception technique * Produire un document d’architecture * Produire un plan de projet et des plans d’itérations agiles ; * Maitriser un IDE. * Concevoir une interface utilisateur en mode scénarisation dans un IDE (Édition graphique générant un XML) ; * Gérer les événements, les routages d’événements et les autres concepts propres à la navigation dans une interface graphique. * Découpler la communication, le modèle de données et l’interface utilisateur ; * Appliquer des tests unitaires aux fonctionnalités des modules découplés. * Utiliser un gestionnaire de source * Intégrer du contenu multilingue * Bien documenter le code * Connaitre le processus de distribution des applications mobiles. |

| **Critères de performance** |
| --- |
| **1. Établir les fonctionnalités de l’application.**   1. Précision des besoins du client ou de la cliente. 2. Collecte complète d’information sur le nombre et le type d’utilisatrices et d’utilisateurs. 3. Collecte complète d’information sur l’ampleur, la nature et le degré d’interactivité de l’application. 4. Collecte complète de données sur le mode de diffusion de l’information. 5. Déduction et justification des fonctionnalités de l’application. 6. Production et présentation d’un rapport complet et clair. |
| **2. Établir le cadre technologique.**   1. Reconnaissances des facteurs influant sur les choix technologiques. 2. Détermination de l’architecture appropriée. 3. Prise en considération des possibilités d’évolution du contexte d’utilisation et des technologies. 4. Choix du matériel et des logiciels appropriés à l’architecture retenue. 5. Estimation réaliste des coûts. 6. Production et présentation d’un rapport complet et clair. |
| **3. Préparer le travail de développement de l’application.**   1. Établissement d’un calendrier des activités réaliste. 2. Utilisation appropriée des méthodes et des outils de planification. 3. Installation des logiciels retenus dans l’environnement de développement. 4. Appropriation de l’environnement de développement de l’application. |
| **4. Produire le prototype de présentation.**   1. Exploitation correcte des possibilités des outils de développement. 2. Établissement approprié des caractéristiques de l’exploration. 3. Représentation graphique correcte des écrans. 4. Détermination des caractéristiques globales de présentation des écrans. 5. Représentation de l’interface de l’application conformément aux exigences de l’ergonomie et de l’esthétisme. 6. Présentation du prototype pour approbation aux personnes responsables. |
| **5. Produire le prototype de communication.**   1. Exploitation correcte des possibilités des outils de développement. 2. Détermination du cheminement de l’information à travers les niveaux. 3. Programmation du squelette de communication entre les niveaux en tenant compte du volume transactionnel et des accès concurrents. 4. Validation du squelette de communication. 5. Adaptation appropriée des choix technologiques. |
| **6. Développer l’application.**  6.1 Codification appropriée des écrans.  6.2 Codification des fonctions de l’application conformément aux exigences de l’entreprise.  6.3 Création et modification appropriées de la base de données.  6.4 Codification correcte des requêtes d’accès à la base de données.  6.5 Codification de l’intégration des différents éléments dans le squelette de communication.  6.6 Vérification rigoureuse du fonctionnement de chacun des programmes et de l’application dans l’environnement de développement.  6.7 Validation et optimisation des performances de l’application.  6.8 Production complète et archivage de toute l’information relative aux programmes. |
| **7. Produire la documentation relative à l’application.**  7.1 Modification appropriée de toute l’information relative à l’application.  7.2 Création appropriée de l’aide en ligne.  7.3 Rédaction claire et complète des instructions d’utilisation de l’application.  7.4 Inscription du site contenant l’application auprès des moteurs de recherche appropriés. |

### CONTENU DU COURS

1. **Introduction à Java** (9 heures)
   1. Révision des notions OO
   2. Librairies de bases
2. **Introduction au système Android** (9 heures)
   1. Particularités des applications en mobilité
   2. Système d’exploitation
   3. Cycle de vie des applications
   4. Senseurs et autres périphériques
3. **Conception d’interfaces utilisateurs flexibles** (15 heures)
   1. Concepts
   2. Composantes graphiques
   3. Alertes et notifications
   4. Grilles et fragments
   5. Couplage de données
   6. Chargement dynamique des contenus
   7. Multilinguisme
4. **Processus, fils d’exécutions et services** (12 heures)
5. **Persistance locale des données** (9 heures)
   1. Types de stockages
   2. Modèles de sécurité
   3. Fichiers
   4. Bases de données locales
6. Utilisation de la géolocalisation (6 heures)
7. Intégration des services externes (6 heures)
8. Aide et documentation intégrée (6 heures)
9. Monétisation et cycles de déploiements (6 heures)

### MÉTHODOLOGIE

La théorie sera présentée à l’aide de nombreux exemples. Afin de consolider les notions vues en classe, des exercices et des travaux pratiques seront exigés. Selon les besoins des étudiants et le temps disponible, le professeur agira comme conseiller pour solutionner les exercices et autres défis en classe.

En laboratoire, l'étudiant sera amené à développer des algorithmes complets ou partiels et à les programmer dans le but d'appliquer les notions théoriques.

L'étudiant sera également amené à modifier des programmes existants selon des spécifications énoncées. De plus, l'étudiant sera appelé à bâtir des applications complètes et de valider leurs fonctionnements avec des plans de tests et des tests automatisés.

Les outils de gestion de sous-groupes, notes de cours, exercices et évaluations formatives sont disponibles sur la plateforme **Moodle** du CEGEP. Les documents de projets que vous produirez seront sur la plateforme collaborative **Google Docs** et le code source sera sur le gestionnaire **GitHub**.

**travaux**

3 Travaux pratiques d’introduction 3x10%

Moment  : Semaines 1 à 6

Mode : Individuel

Tâche  : Production guidée d’un logiciel Android basé sur du code existant et des spécifications techniques détaillées.

Durée  : 2 cours, réalisable en classe.

Projet 1 10%

Moment  : Semaine 7 à 10

Mode : En équipe de 2 ou 3

Tâche  : Production collaborative d’un logiciel Android selon les spécifications fournies, document de conception et tenue d’un journal de bord. Les étudiants doivent planifier et faire approuver des sous-tâches de moins de 2 heures et procéder à la réalisation.

Durée  : 6 cours

Projet 2 15%

Moment  : Semaine 11 à 15

Mode : Équipes de 3 à 5

Tâche  : Production collaborative d’un logiciel Android, document de conception et tenue d’un journal de bord collectif. L’équipe doit planifier le projet, définir et faire approuver des sous-tâches de moins de 2 heures et procéder à la réalisation.

Durée  : 11 cours

**Notes :**

1. Les notes des travaux d’équipes sont **individualisées** en fonction du résultat combiné des *Fiches d’évaluations du travail d’équipe*. Il s’agit d’une évaluation en 15 points de l’attitude de chacun des membres de l’équipe. **Les notes individualisées peuvent varier considérablement au sein d’une même équipe**.
2. Les travaux de documentation doivent être rédigés conformément au *Guide de présentation d’un travail écrit*. Cette rédaction doit être faite en dehors du temps de classe afin de maximiser la pratique supervisée.

**Examens (Évaluations sommatives)**

Examen intra :

Pondération : 15%  
 Moment  : jeudi 6 octobre 2016

Mode : Individuel

Tâche  : Questionnaire suivi de la production d’un logiciel simple basé sur du code existant et des spécifications techniques.

Durée  : 3 heures

Examen final :

Pondération : 30%

Moment  : lundi 12 décembre 2016

Mode : Individuel

Tâche  : Questionnaire suivi de la production d’une portion de logiciel basé sur du code existant et des spécifications techniques.

Durée  : 3 heures

**Note** La somme de ces 2 évaluations constitue l’**épreuve terminale du cours**, soit 45% de la note finale. Toutes les prescriptions de la PDEA et de la PIEA seront appliquées telles quelles, en particulier une moyenne de 60% pour les examens ainsi que pour les travaux est exigée. La réussite des travaux est aussi essentielle que celle des examens.

Un maximum de 10% des points sera prélevé sur les travaux ou examens pour la qualité du français.

### MÉDIAGRAPHIE

Ce cours utilisera de nombreux documents de référence et exemples accessibles gratuitement sur Internet. Aucun achat n’est requis. La liste des ouvrages et leurs liens de téléchargements sont à l’adresse suivante : <http://curomediares.com/tag/420-gep-hy>

**RÈGLES PARTICULIÈRES**

Les règlements s’appliquant aux laboratoires informatiques.

Tout étudiant(e) qui s'absente a l'obligation de s'informer auprès de ses confrères de classe à propos de la matière présentée, des documents remis par le professeur, des travaux à accomplir et des évaluations qui ont été tenues.

Une absence non motivée à une évaluation équivaut à un 0.

Une absence motivée permet un report de l'évaluation.