

**TECHNOLOGIES ET INSTRUMENTS DE DEVELOPPEMENT DES PRODUITS LOGICIEL**

**1. Informations sur l'unité de cours / module**

<b>Faculté</b>	Ordinateurs, Informatique et Microélectronique				
<b>Chaire/département</b>	Filière Francophone Informatique, dép. Génie Logiciel et Automatique				
<b>Cycle d'études</b>	Études supérieures, Licence - cycle I				
<b>Programme d'études</b>	526.2 Technologies de l'information				
<b>Année d'étude</b>	<b>Semestre</b>	<b>Type d'évaluation</b>	<b>Catégorie formative</b>	<b>Catégorie d'option</b>	<b>Crédites ECTS</b>
II (enseignement à plein temps);	5	E	S – Unité de cours de spécialité	O - Unité de cours obligatoire	5

**2. Estimation du temps total**

Nombre total d'heures dans le programme	Dont				
	Heures dans la salle de cours		Travail individuel		
	Cours	Travaux pratique/dirigés	Projet d'année	Étude du matériel théorique	Préparation de l'application
90	45	30/0	0	45	30

**3. Prérequis pour l'accès à l'unité de cours/module**

Selon le programme d'études	Bases de la programmation en Java ou en C#. Connaître les instructions de base pour utiliser les bases de données. Utilisation d'Internet.
Selon les compétences	Connaissances et compétences pour développer des applications informatiques, interface graphique, etc.

**4. Conditions de déploiement le processus éducatif pour**

Cours	Pour présenter le matériel théorique en classe, il faut un tableau noir, un projecteur et un ordinateur.
Travaux pratique/dirigés	Les étudiants rédigeront des rapports selon les conditions formulées dans les indications méthodiques. La durée du soutien d'un travail pratique est une semaine après l'achèvement.

**5. Compétences spécifiques accumulées**

Compétences professionnelles	<ul style="list-style-type: none"> <li>✓ Notions générales sur les techniques et les outils pour le développement de logiciel. Techniques et types d'architectures. Système de contrôle de version et travail collaboratif. Les tendances de développement des outils de programmation. Types d'outils CASE.</li> <li>✓ Outils de modélisation. Méthodes formelles. Classification formelle. Outils pour l'assurance de la qualité.</li> <li>✓ Intégration continue, livraison continue. Comprendre, connaître et utiliser des outils de construction automatisés.</li> <li>✓ Aspects de sécurité pour le développement d'une application Web;</li> <li>✓ Utilisation de CMS et de cadres de développement d'applications Web; utilisation des bibliothèques pour faciliter la mise en œuvre;</li> </ul>
------------------------------	--

Compétences	CT1. Appliquer les principes, les normes et les valeurs de l'éthique professionnelle.
-------------	---

transversales	
	CT2. Identifier, décrire et gérer les activités organisées en équipe ; développement des capacités de communication et de collaboration.

### 6. Objectifs de l'unité de cours / module

<b>Objectif général</b>	Développer des compétences pour le développement de produits logiciel modernes, en fonction des besoins actuels du domaine
<b>Objectifs spécifiques</b>	Acquérir des compétences pour concevoir et développer des produits logiciel. Sélectionnez efficacement la technologie la mieux adaptée à un produit logiciel. Utiliser des techniques pour concevoir et développer des programmes avec des instruments modernes en respectant les critères de qualité en vigueur.

### 7. Contenu de l'unité de cours / module

Thématique des activités didactiques	Nombre d'heures	
	enseignement à temps plein	enseignement à temps partiel
Thème des cours		
<b>T1. INTRODUCTION</b> Concepts généraux sur les techniques et les outils de développement des logiciels. Techniques et types d'architectures. Système de contrôle de version et travail collaboratif. Les tendances de développement des outils de programmation. Types d'outils CASE.	12	
<b>T2. Outils de modélisation et d'assurance qualité.</b> Outils de modélisation. Méthodes formelles. Classification formelle. Outils d'assurance qualité: refactoring, modularité, réflexion. Types et formes de programmation visuelle. Paradigmes de la programmation visuelle.	8	
<b>T3. Outils de développement (IDE)</b> Outils de développement créés par Jet Brains. Utilisation d'IntelliJ IDEA pour développer des programmes en JAVA. Utiliser les tests unitaires, l'injection de dépendance et les maquettes. Utiliser le débogueur, l'éditeur de liens et le traducteur. Paradigme de développement logiciel rapide	8	
<b>T4. Intégration continue, livraison continue.</b> Utilisation de l'environnement logiciel de gestion de projet (JIRA). Intégration continue, livraison continue. Comprendre, connaître et utiliser des outils de construction automatisés (Ant, Maven, Grandle). Tests de performance. Profiling. Virtualisation, conteneurs. Docker.	9	
<b>T5. Applications Web et services Web.</b> Applications Web et services Web. Utilisation de Bootstrap pour implémenter l'interface Web. JQuery. Utilisation de CMS et de cadres pour faciliter le développement d'applications Web.	8	
Total des cours:	45	

Thématique des activités didactiques	Nombre d'heures	
	enseignement à temps plein	enseignement à temps partiel
Thèmes des travaux pratiques		
TP1 Étude du système de gestion de versions Git et des commandes de base de Git Bash. Création d'un compte sur le serveur Git. Copier à partir de et sur le serveur Git. Partage de ressources avec le groupe de travail.	4	
TP2 Explorez les possibilités de l'IDE Eclipse. Utilisez des outils de base. Installez et utilisez des extensions supplémentaires. Développer une application en utilisant Eclipse.	4	
TP3 Utilisez les tests unitaires, l'injection de dépendance et les maquettes.	4	
TP4 Utilisez le débogueur, l'éditeur de liens html et le traducteur sur Eclipse IDE et/ou IntelliJ IDEA.	4	
TP5 Génération automatique et documentation du code sur les IDE étudiés.	4	
TP6 Utilisez un CMS ou un framework pour développer une application Web.	6	
Total des travaux pratiques:	30	

### 8. Références bibliographiques

Principales	<ol style="list-style-type: none"> <li>1. Kent Beck. <i>Extreme Programming Explained: Embrace Change</i>. Addison-Wesley, Boston , 2005</li> <li>2. Martin Fowler. <i>Refactoring - Improving the Design of Existing Code</i>. Addison-Wesley, Boston , 1999.</li> <li>3. Ken Auer, Roy Miller. <i>Extreme Programming Applied: Playing to Win</i>. Addison-Wesley, Boston , 2002.</li> <li>4. Alec Sharp. <i>Workflow Modeling: Tools for Process Improvement and Application Development</i>. Artech Hous, Boston , 2009.</li> <li>5. Frank W. Liou. <i>Rapid Prototyping and Engineering Applications: A Toolbox for Prototype Development</i>. CRC Press, 2008</li> <li>6. Jami Clott, Stephen Raff. <i>Peoplesoft Application Development Tools</i>. McGraw-Hill, 2000</li> <li>7. Dinesh Rajput. <i>Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more</i>. Packt Publishing, Birmingham, 2017</li> </ol>
Supplémentaires	<ol style="list-style-type: none"> <li>1. <a href="http://www.testdriven.com">http://www.testdriven.com</a></li> <li>2. <a href="http://www.xprogramming.com">http://www.xprogramming.com</a></li> <li>3. <a href="http://www.refactoring.com">http://www.refactoring.com</a></li> <li>4. <a href="http://www.pairprogramming.com">http://www.pairprogramming.com</a></li> <li>5. <a href="http://AgileAlliance.com">http://AgileAlliance.com</a></li> <li>6. <a href="http://ExtremeProgramming.org">http://ExtremeProgramming.org</a></li> <li>7. <a href="http://www.junit.org">http://www.junit.org</a></li> </ol>

Note semestrielle				Moyenne tronquée (0.6)	Examen final (0.4)
Composants					
Evaluations périodiques		Evaluations courant	Travail individuel		
Attestation 1	Attestation 2				
Normes de rendement minimum					
Présence et activité aux cours et travaux pratiques; Obtenez le score minimal de "5" pour chacune des attestations et des travaux pratiques; Démonstration de l'assimilation des informations fournies pendant le cours et des compétences.					