

PROGRAMMATION ORIENTÉE OBJET
1. Données sur la discipline

| | | | | | |
|---|--|--------------------------|--|--------------------------------|---------------------|
| Faculté | Ordinateurs, informatique et microélectronique | | | | |
| Département | Génie Logiciel et Automatique | | | | |
| Cycle des études | Études supérieures de premier cycle, cycle I | | | | |
| Programme d'études | 0613.1 Technologie d'information | | | | |
| L'année d'études | Semestre | Type d'évaluation | Catégorie formative | Catégorie facultative | Crédits ECTS |
| L'année II (<i>éducation à plein temps</i>) | III | E | D-Discipline du domaine professionnel. | O - unité de cours obligatoire | 6 |
| L'année II (<i>éducation à temps partiel</i>) | IV | E | D-Discipline du domaine professionnel. | O - unité de cours obligatoire | 6 |

2. Durée totale estimée

| Nombre total d'heures dans le programme | Dont | | | | |
|---|---------------------------|-----------------------|--------------------|-------------------------------|----------------------------|
| | Heures de cours en classe | | Travail individuel | | |
| | Cours | Laboratoire/séminaire | Projet de l'année | L'étude du matériel théorique | Préparation d'applications |
| Éducation à plein temps | 30 | 30/30 | | 90 | |
| Éducation à temps partiel | 12 | 12/12 | | 144 | |

3. Conditions préalables pour accéder à la discipline

| | |
|----------------------------------|--|
| Selon le plan d'éducation | G.O.001 Programmation des ordinateurs F.O Structures de données et algorithmes |
| Selon les compétences | Connaissance et compétences en matière de conception et de développement d'algorithmes et de programmes en langage C pour résoudre des problèmes informatiques |

4. Conditions de réalisation du processus éducatif pour

| | |
|------------------------------|---|
| Cours | Pour la présentation du matériel théorique en classe sont nécessaires: un projecteur, un PC/ordinateur portable et un accès Internet. Les retards des étudiants et les appels téléphoniques pendant les cours ne seront pas tolérés |
| Laboratoire/séminaire | Les étudiants complèteront des rapports selon les conditions formulées dans les consignes méthodiques. La date limite pour soutenir les travaux de laboratoire est une semaine après leur achèvement. |

5. Compétences spécifiques accumulées

| | |
|-------------------------------------|---|
| Compétences professionnelles | CP2. Conception et développement d'applications CP7. Ingénierie des systèmes |
| Compétences transversales | CT1. Appliquer les principes, normes et valeurs de l'éthique professionnelle. CT2. Identifier, décrire et réaliser les activités organisées en équipe avec le développement des capacités de communication et de collaboration, mais aussi avec la prise en charge de différents rôles (d'exécution et de gestion). CT3. Faire preuve d'un esprit d'initiative et d'action pour mettre à jour ses propres connaissances en matière de culture professionnelle, économique et organisationnelle. |

6. Objectifs de la discipline

| | |
|---------------------------|--|
| L'objectif général | L'acquisition par les étudiants de notions, concepts et exemples issus de la programmation orientée objet, Familiarisation des étudiants avec les techniques de base spécifiques à la programmation orientée objet, Construction et analyse d'algorithmes spécifiques à la |
|---------------------------|--|

| | |
|----------------------------------|---|
| | programmation orientée objet. |
| Les objectifs spécifiques | Une fois cette discipline terminée avec succès, les étudiants seront capables de: utiliser les techniques de programmation orientée objet (POO); implémenter des techniques de POO en C++ ; créer des applications basées sur la POO. |

7. Le contenu de la discipline

| Thématique des activités didactiques | Nombre d'heures | |
|--|-------------------------|---------------------------|
| | éducation à plein temps | éducation à temps partiel |
| Thématique des cours | | |
| T1. POO Paradigmes de programmation. Concepts de base de la POO | 2 | 1 |
| T 2. Facilités C++ | 2 | |
| T 3. Définition des classes. Fonctions spéciales Constructeur et destructeur | 2 | 1 |
| T 4. Particularités: classes et objets. Fonctions et classes friend | 2 | 1 |
| T5-6. Surcharge des opérateurs | 4 | 1 |
| T 7. Héritage. Dérivation de classe simple | 2 | 1 |
| T 8. Fonctions virtuelles et polymorphisme | 2 | 1 |
| T 9. Héritage. Dérivation multiple de classes | 2 | 1 |
| T 10. Hiérarchies de classes | 2 | 1 |
| T 11. Programmation générique | 2 | 1 |
| T 12-13. Standard Template Library (STL) | 4 | 1 |
| T 14. Traitement des exceptions | 2 | 1 |
| T 15. Exemples de mise en œuvre des concepts POO | 2 | 1 |
| Total cours: | 30 | 12 |
| Thématique des travaux pratiques/séminaires | | |
| LP1. Structure – mécanisme d'abstraction | 2 | 1 |
| LP2. Facilités du langage de programmation C++ par rapport au langage C | 2 | |
| LP3. Classes et objets. Constructeurs et destructeurs pour la classe | 2 | 1 |
| LP4. Fonctions et classes friend. Utilisation de *this et static dans le développement de programmes dans le style orienté objet | 2 | 1 |
| LP5. Surcharge d'opérateurs unaires et binaires | 4 | 1 |
| LP6. Réaliser un héritage simple entre deux classes ou plus | 2 | 1 |
| LP7. Polymorphisme. Fonctions virtuelles. Cours virtuels | 2 | 1 |
| LP8. Héritage multiple. Cours abstraits. Ambiguïtés dans la relation d'héritage multiple | 2 | 1 |
| LP9. Relations entre les classes. Association. Agrégation. La composition | 2 | 1 |
| LP10. Modèles de fonctions et modèles de classes. | 2 | 1 |
| LP11. Conteneurs. Itérateurs. Algorithmes | 4 | 1 |
| LP12. Gestion des exceptions prédéfinies | 2 | 1 |
| LP13. Implémentation des concepts POO | 2 | 1 |
| Total travaux pratiques/séminaires: | 30 | 12 |
| Thématique des travaux de laboratoire | | |
| LL1. Structure – mécanisme d'abstraction | 2 | 1 |
| LL2. Facilités du langage de programmation C++ par rapport au langage C | 2 | |
| LL3. Classes et objets. Constructeurs et destructeurs pour la classe | 2 | 1 |
| LL4. Fonctions et classes friend. Utilisation de *this et static dans le développement de programmes dans le style orienté objet | 2 | 1 |
| LL5. Surcharge d'opérateurs unaires et binaires | 4 | 1 |
| LL6. Réaliser un héritage simple entre deux classes ou plus | 2 | 1 |
| LL7. Polymorphisme. Fonctions virtuelles. Cours virtuels | 2 | 1 |
| LL8. Héritage multiple. Cours abstraits. Ambiguïtés dans la relation d'héritage multiple | 2 | 1 |
| LL9. Relations entre les classes. Association. Agrégation. La composition | 2 | 1 |
| LL10. Modèles de fonctions et modèles de classes. | 2 | 1 |

| Thématique des activités didactiques | Nombre d'heures | |
|---|-------------------------|---------------------------|
| | éducation à plein temps | éducation à temps partiel |
| LL11. Conteneurs. Itérateurs. Algorithmes | 4 | 1 |
| LL12. Gestion des exceptions prédéfinies | 2 | 1 |
| LL13. Implémentation des concepts POO | 2 | 1 |
| Total travaux de laboratoire: | 30 | 12 |

8. Références bibliographiques

| | |
|-----------------------|---|
| Principales | <ol style="list-style-type: none"> 1. Microsoft Developer Network (MSDN) 2. Herbert Schildt, C++, Ed. Teora (traducere, 2002); 3. Kris Jamsa si Lars Klander, Totul despre C si C++ Manualul fundamental de programare in C si C++, Ed. Teora, (traducere 2007); 4. David Vandevorde, Nicolai M. Josuttis "C++ Templates: The Complete Guide". Addison Wesley, 2002 5. Heileman Gregory L. Data Structures, Algorithms and Object Oriented Programming. – McGraw – Hill, 1996. |
| Suplimentaires | <ol style="list-style-type: none"> 1. Bruce Eckel "Thinking in C++", 2000 2. Bartosz Milewski "C++ In Action. Industrial-Strength Programming Techniques" 3. Jeffrey Richter, Applied Microsoft .NET Framework Programming, Microsoft Press (2002) 4. Gîncu S. Metodologia rezolvării problemelor de informatică în stilul orientat pe obiecte, Chişinău, 2012, 112 p. |

9. Évaluation

| Périodique | | Courante | Etude individuelle | Projet/thèse | Examen |
|----------------------------------|------|----------|--------------------|--------------|--------|
| EP 1 | EP 2 | | | | |
| Éducation à plein temps | | | | | |
| 15% | 15% | 15% | 15% | | 40% |
| Éducation à temps partiel | | | | | |
| 25% | | | 25% | | 50% |
| Standard de performance minimale | | | | | |

10. Critères d'évaluation

| Activité | Composantes d'évaluation | Méthode d'évaluation, Critères d'évaluation | Poids dans la note finale de l'activité | Le poids dans l'évaluation de la discipline |
|----------------------------------|---------------------------------|---|---|---|
| Éducation à plein temps | | | | |
| Évaluation périodique I | Contenu théorique, thèmes 1-6 | Test sur MOODLE | 100% | 15% |
| Évaluation périodique II | Contenu théorique, thèmes 7-12 | Test sur MOODLE | 100% | 15% |
| Évaluation courante | Activité pratique | Discussions en séminaires | 50% | 15% |
| | | Dossier complété avec les rapports | 50% | |
| Etude individuelle | Implémentation des concepts POO | Produits du programme développés et présentés | 100% | 15% |
| Évaluation finale | Contenu théorique et pratique | Épreuve écrite. Marquage selon l'échelle | 100% | 40% |
| Éducation à temps partiel | | | | |
| Évaluation périodique I | Contenu théorique, thèmes 1-6 | Test sur MOODLE | 40% | 25% |
| Évaluation périodique II | Contenu théorique, thèmes 7-12 | Test sur MOODLE | 40% | |
| Évaluation courante | Activité pratique | Dossier complété avec les rapports | 20% | |
| | Implémentation des concepts POO | Produits du programme développés et présentés | 100% | 25% |

| Activité | Composantes d'évaluation | Méthode d'évaluation, Critères d'évaluation | Poids dans la note finale de l'activité | Le poids dans l'évaluation de la discipline |
|---------------------------|-------------------------------|---|---|---|
| Etude individuelle | Contenu théorique et pratique | Épreuve écrite. Marquage selon l'échelle | 100% | 50% |