**Fichedescriptived’uneunitéd’enseignement(UE)etdesesélémentsconstitutifs(ECUE)**

**Intitulé de l’UE : Programmation et compilation**

|  |
| --- |
| **Nombredecrédits: 6** |
| **CodeUE: Programmation et compilation** |

|  |  |
| --- | --- |
| **Université: Université Tunis El Manar** | **Établissement : Institut supérieur d’Informatique** |

|  |  |  |
| --- | --- | --- |
| **Domainedeformation: Informatique** | **Mention: Computer Engineering /** **Computer Science** | |
| **Diplôme: ingénieur**  **Parcours**:TroncCommun | | **Semestre: 2** |

1. **Prérequis**(définirlesUEetlescompétencesindispensablespoursuivrel’UEconcernée)

|  |
| --- |
| * Les notions de base sur les systèmes d’exploitation / La gestion des périphériques / La gestion du Système de fichier / Environnement Linux * Algorithme et structure des données / Programmation / Théorie des langages et automates |

**2- Objectifs de l’UE** (utiliser la taxonomie de Bloom ou autre référence reconnue, les objectifs peu-ventêtrespécifiésicioubien au niveaude chaqueECUE,ils doivent êtreconformes)

|  |
| --- |
|  |

**3- Élémentsconstitutifsdel’UE(ECUE)**

* 1. **Élémentsconstitutifsdel’UE**(ECUE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Élémentsconstitutifs | **Volumehoraire(1semestre=14semaines)** | | | | **Crédits** |
|  | **Cours** | **TD** | **TP** | **Autres** |
| Programmation objet | 30 |  | 15 |  | 3 |
| Techniques de compilation | 22,5 | 15 | 7.5 |  | 3 |
| **Total** | 52.5 | 15 | 22.5 |  | 6 |

* 1. **Activités pratiques (Projets, stages, mémoires, …)**

|  |
| --- |
| * 1. Des activités pratiques de cette UE sont en relation avec les techniques abordées dans les cours des deux ECUE. |

**4- Contenu**(descriptifsetplans descours)

**Enseignements**: voir les fiches modules détaillées de chaque ECUE

|  |
| --- |
| Cette Unité d’enseignement permet aux apprenants de développer leurs compétences dans les domaines de la compilation et des systèmes d’exploitation. |

**Activitéspratiquesdel’UE**

|  |
| --- |
| Des séances de travaux pratiques dans l’ECUE systèmes d’exploitation2, sont programmées pour expérimenter les notions de système d’exploitation abordées en cours. |

**5- Méthodes pédagogiques et moyens didactiques spécifiques à l’UE** (méthodes et outils pédagogiques, ouvrages deréférence, recoursauxTIC/possibilités d’enseignementàdistance, ...)

|  |
| --- |
| 1. Support de cours détaillé de chaque chapitre du cours 2. Des références bibliographiques (livres, lien web) sont à fournie aux apprenants 3. Des outils appropriés aux contenus de chaque modules seront installés, paramétrés et utilisés lors des séances des activités pratiques. |

**5- Examensetévaluationdesconnaissances**

**Méthodesd’évaluationetrégimed’examens**(Présenterlerégimed’évaluationpréconisé :contrôlecontinuuniquementou,régimemixtec.à.d.contrôlecontinue etexamensfinaux)

|  |
| --- |
| 1. **Régimemixte:** 2. **Régimemixte:** |

**Validation de l’UE** (préciser les poids des épreuves d’examens pour le calcul de la moyenne del’UCUE,lescoefficients desECUE et lecoefficientde l’UEauseinduparcours)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ECUE** | **Contrôlecontinue** | | | | **ExamenFinal** | | | | **Coef.Del’ECUE** | **Coef.Del’UEauseinduparcours** |
| **Epreuves** | | | **Pondération** | **Epreuves** | | | **Pondération** |
| **Ecrit** | **Oral** | **TP etAutres** | **Ecrit** | **Oral** | **TP etAutres** |
| Programmation objet | x |  | x | 30% | X |  |  | 70% | 2 | 4 |
| Techniques de compilation | x |  | x | 30% | X |  |  | 70% | 1 |

Fichedescriptivedel’ECUE

**Unité d’Enseignement : Programmation et compilation**

**ECUE no 1.1 : Programmation objet**

Plandel’élémentconstitutif(ECUE)

***Programmation Orientée objet***

Chapitre 1 : Les nouvelles possibilités de C++

* Les nouveaux types
* Les entrées/sorties
* Conversion des types en C++
* Les arguments par défauts
* Surcharge
* Notion de référence

Chapitre 2 : L’approche Orientée Objet

* Pourquoi utiliser la programmation orientée objet
* Type de données Abstrait vers l’Orienté Objet
* Le principe d’encapsulation
* La notion classe (attributs, méthodes)

Chapitre 3 : Classe et Objet en C++

* Structure d’un programme C++
* Classe, attributs et méthodes en C++
* Instanciation des classes
* Les droits d’accès /fonctions amies
* Les getters/setters
* Attributs dynamiques /attributs statiques
* Constructeur/destructeur et constructeur par recopie
* Objet membre
* Tableaux d’objets

Chapitre 4 : Surcharge des opérateurs

* Notion de surcharge
* Surcharge avec des fonctions amies
* Surcharge avec des fonctions membres

Chapitre 5 : Héritage et Polymorphisme

* Notion d’héritage
* Règles d’accès et héritage
* Redéfinition des méthodes
* Fonctions virtuelles
* Fonctions virtuelles pures
* Classes abstraites

Chapitre 6 : Généricité

* Notion de généricité
* Template de fonction
* Template de classe

Chapitre 7 : Les conteneurs STL

* Bibliothèque standard de C++
* Les vecteurs
* Les listes
* Les itérateurs

Prérequis:

|  |
| --- |
| 1. Algorithmique et structure de données 2. Atelier de programmation 1 et 2 |

**Objectifsd’apprentissagedel’élémentconstitutif(ECUE) :**

|  |
| --- |
| Au terme de l’élément constitutif, l’étudiant sera en mesure de :  (Spécifier les objectifs l’élément constitutif (OE) en utilisant des verbes selon la taxonomie de Bloom ou autre taxonomie de référence, maximum 8)  OE1: Comprendre la Programmation Orientée Objet (POO)  OE2: Comprendre les spécificités du langage C++  OE3: analyser, concevoir et résoudre des problèmes selon l’approche OO. |

Matrice d’association entre les objectifs de l’élément constitutif et ceux du programmedeformation(OP)

Parcours : Ingénierie des Systèmes Embarqués et Objets Connectés (ISEOC)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** |
| **CS2** | X | X | X |
| **CS3** | X | X | X |
| **CS5** | X | X | X |
| **CS8** | X | X | X |

Parcours : Ingénierie et Développement des Infrastructures et des Services de Communications (IDISC)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** |
| **CS1** | X | X | X |
| **CS3** | X | X | X |
| **CS4** | X | X | X |

Parcours : Ingénierie du Développement du Logiciel (IDL)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** |
| **CS2** | X | X | X |
| **CS3** | X | X | X |
| **CS4** | X | X | X |
| **CS5** | X | X | X |
| **CS7** | X | X | X |

Descriptiondecontenul’élémentconstitutif

Spécifier les chapitres et leurs sous sections en adoptant l’organisation d’un livre de référenceconnu et référée dans le domaine. Spécifie également le nombre d’heures de cours (C), CI,TD,TP, projet ou autresnécessaires pour chaquechapitreou partie.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapitreetsections** | **Cours(h)** | **CI (h)** | **TD(h)** | **TP (h)** | **Projet(h)** | **Total(h)** |
| **Chapitre1** | x |  |  | x |  |  |
| **Chapitre2** | x |  |  | x |  |  |
| **Chapitre3** | x |  |  | x |  |  |
| **Chapitre 4** | x |  |  | x |  |  |
| **Chapitre 5** | x |  |  | x |  |  |
| **Chapitre 6** | x |  |  | x |  |  |
| **Chapitre 7** | x |  |  | x |  |  |

Livresderéférences:

1. Ellis, M. A., & Stroustrup, B. (1990). The annotated C++ reference manual. Addison- Wesley

Musser, D. R., Derge, G. J., & Saini, A. (2009). STL tutorial and reference guide : C++ programming with the standard template library. Addison-Wesley Professional.

Fichedescriptivedel’ECUE : Techniques de compilation

**Unité d’Enseignement : Programmation et compilation**

**ECUE no 1.2 : Techniques de compilation**

**Plan de l’élément constitutif (ECUE) :**

|  |
| --- |
| **Plan du cours** |
| **Chapitre 1 :** Introduction à la compilation  **Chapitre 2 :** Analyse Lexicale  1. Rôle de l'analyseur lexical  2. Spécification des unités lexicales  3. Traitement du code source  4. Analyse lexicale vs. Automates finis  5. Implémentation d'un analyseur lexical  6. Gestion des erreurs  7. Utilisation de lex / flex  **Chapitre 3 :** Analyse Syntaxique  1. Introduction  2. Rôle de l'analyseur syntaxique  3. Ecriture d'une grammaire  4. Grammaire ambigüe  5. Analyse descendante  6. Analyse ascendante  7. Analyse par précédence d'opérateurs  8. Analyseurs LR  9. Utilisation de grammaires ambigües  10. Constructeurs d'analyseurs syntaxiques yacc / bison  **Chapitre 4 :** Traduction dirigée par la syntaxe  1. Les définitions dirigées par la syntaxe  2. Construction des arbres abstraits  3. Evaluation ascendante des définitions S-Attribuées  4. Définitions L-Attribuées  5. Traduction descendante  6. Evaluation ascendante des attributs hérités  7. Evaluateurs récursifs  8. Analyse des définitions dirigées par la syntaxe  **Chapitre 5 :** Contrôle de types  1. Système de typage  2. Contrôleurs de types  3. Equivalence des expressions de types  4. Conversions de types  5. Surcharge des fonctions et des opérateurs  6. Polymorphisme de fonction  **Chapitre 6 :** Génération de code  1. Production de code intermédiaire  2. Production de code  3. Optimisation de code |

Prérequis:

|  |
| --- |
| * Algorithme et structure des données * Programmation * Théorie des langages et automates |

**Objectifsd’apprentissagedel’élémentconstitutif(ECUE) :**

|  |
| --- |
| A la fin de ce cours les étudiants seront en mesure de :  OE1 : Maitriser le fonctionnement global et les différentes étapes d’un compilateur.  OE2 : Maitriser les détails de l’analyseur lexicale et syntaxique.  OE3 : Manipuler les définitions dirigées par la syntaxe et le contrôle de type.  OE4 : Manipuler des outils permettant de développer des compilateurs (flex, bison, …). |

Matrice d’association entre les objectifs de l’élément constitutif et ceux du programmedeformation(OP)

Parcours : Ingénierie des Systèmes Embarqués et Objets Connectés (ISEOC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** | **OE4** |
| **CS2** | X | X | X | X |
| **CS3** | X | X | X | X |
| **CS5** | X | X | X | X |
| **CS8** | X | X | X | X |

Parcours : Ingénierie et Développement des Infrastructures et des Services de Communications (IDISC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** | **OE4** |
| **CS1** | X | X | X | X |
| **CS3** | X | X | X | X |
| **CS4** | X | X | X | X |

Parcours : Ingénierie du Développement du Logiciel (IDL)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **OE1** | **OE2** | **OE3** | **OE4** |
| **CS2** | X | X | X | X |
| **CS3** | X | X | X | X |
| **CS4** | X | X | X | X |
| **CS5** | X | X | X | X |
| **CS7** | X | X | X | X |

**Descriptionducontenude l’élémentconstitutif**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chapitreetsections** | **Cours(h)** | **CI (h)** | **TD(h)** | **TP (h)** | **Projet(h)** | **Total(h)** |
| **Chapitre1** | X |  | X | X |  |  |
| **Chapitre2** | X |  | X | X |  |  |
| **Chapitre3** | X |  | X | X |  |  |
| **Chapitre4** | X |  | X | X |  |  |
| **Chapitre5** | X |  | X | X |  |  |
| **Chapitre6** | X |  | X | X |  |  |

Livresderéférences:

|  |
| --- |
| 1. Reghizzi, Stefano Crespi, Luca Breveglieri, and Angelo Morzenti. Formal languages and compilation. Springer, 2013.‏  2. J.E. Hopcroft, J.D. Ullman. Introduction to automata theory, languages and computation. Addison-Wesley, 1979.  3. M. Sipser. Introduction to the theory of computation. PWS Publishing Company, 1996.  4. Lingas, R. Karlsson, S. Carlsson. Automata, Languages and Programming. Lecture Notes in Computer Science – 20th International Colluquium ICALP93. Springer- Verlag Ed.  5. Introduction à la calculabilité, P. Wolper, Dunod 2006 (3ème édition).  6. Théorie des automates (méthodes et exercices corrigés), P. Séébold, Vuibert 1999.  7. Méthodes mathématiques pour l’informatique (4ème édition), J. Vélu, Dunod 2005.  8. Théorie des langages et des automates, J.-M. Autebert, Masson 1994.  9. Éléments de théorie des automates, J. Sakarovitch, Vuibert 2003. |