

Progression Term / 2021 - 2022

Développer les compétences suivantes :

- Analyser et modéliser un problème en termes de flux et de traitement d'informations ;
- Décomposer un problème en sous-problèmes, reconnaître et réutiliser des situations ;
- Concevoir des solutions algorithmiques ;
- Traduire un algorithme dans un langage de programmation (interfaces, interactions, comprendre et réutiliser des codes sources existants, développer des processus de mise au point et de validation de programmes) ;
- Mobiliser les concepts et les technologies utiles pour assurer les fonctions d'acquisition, de mémorisation, de traitement et de diffusion des informations ;
- Développer des capacités d'abstraction et de généralisation.

Développer des compétences transversales :

- Faire preuve d'autonomie, d'initiative et de créativité ;
- Présenter un problème ou sa solution, développer une argumentation ;
- Coopérer au sein d'une équipe dans le cadre d'un projet ;
- Rechercher de l'information, partager des ressources ;
- Faire un usage responsable et critique de l'informatique.

1 Langage et programmation - Rappels Python - PJ

1. Python : Rappels - Fonctions et appels de fonction
2. Modularité - Utilisation de bibliothèques ou des API
3. Exploiter la documentation.
4. Rappel - Mise au point de programmes (utiliser des jeux de tests - Doctest)
5. Créer des modules simples et les documenter.

2 Langages et programmation - Mise au point des programmes et gestion des bugs - PR

1. Savoir répondre aux problèmes liés au typage
2. Savoir répondre aux effets de bord non désirés, débordements dans les tableaux
3. Savoir répondre aux instruction conditionnelle non exhaustive
4. Savoir répondre aux choix des inégalités, comparaisons et calculs entre flottants
5. Savoir répondre aux mauvais nommage des variables
6. Rappel - Mise au point de programmes (utiliser des jeux de tests – assert, ...)
7. Comprendre que la calculabilité ne dépend pas du langage de programmation utilisé.
8. Montrer, sans formalisme théorique, que le problème de l'arrêt est indécidable.

3 Structures de données – Programmation objet et autres paradigmes de programmation - PJ

1. Distinguer sur des exemples les paradigmes impératif, fonctionnel et objet.
2. Choisir le paradigme de programmation selon le champ d'application d'un programme.
3. Écrire la définition d'une classe
4. Accéder aux attributs d'une classe
5. Accéder aux méthodes d'une classe
6. Distinguer sur des exemples les paradigmes impératif, fonctionnel et objet.
7. Choisir le paradigme de programmation selon le champ d'application d'un programme.

4 Architectures matérielles OS et réseaux - Protocoles de routage - PR

1. Les tables de routage étant données, identifier la route empruntée par un paquet

2. Calcul du nombre de sauts (protocole RIP)
3. Calcul du coût des routes (protocole OSPF)

=> BAC BLANC 1 (3h 30 min)

=> TP NOTE (1 heure)

5 Langage et programmation – Programme en tant que donnée / Récursivité - PJ

1. Comprendre que tout programme est aussi une donnée.
2. Écrire un programme récursif
3. Analyser le fonctionnement d'un programme récursif
4. Savoir répondre aux effets de bord non désirés

6 Bases de données - Modèle relationnel - PR

1. Relation entre données / Attribut d'une donnée / Domaine / Clef primaire / clef étrangère / schéma relationnel
2. Savoir distinguer la structure d'une base de données de son contenu.
3. Repérer des anomalies dans le schéma (redondances de données, anomalies d'insertion, de suppression, de mise à jour)

7 Structures de données – Structures linéaires : Listes, piles, files - PJ

1. Distinguer Listes, Piles et Files par le jeu des méthodes qui les caractérisent.
2. Distinguer les modes LIFO et FIFO des Piles et des Files
3. Choisir une structure de données adaptée à la situation à modéliser.
4. Distinguer la recherche d'une valeur dans une liste et dans un dictionnaire.
5. Spécifier une structure de données par son interface.
6. Distinguer interface et implémentation.
7. Écrire plusieurs implémentations d'une même structure de données.

8 Structures de données / Algorithmique - Arbre binaire / Arbre binaire de recherche - PR

1. Identifier des situations nécessitant une structure de données arborescente.
2. Définir un arbre en tant que structure hiérarchique (nœuds, racines, feuilles, sous-arbres).
3. Évaluer la taille et la hauteur d'un arbre.
4. Distinguer les arbres binaires
5. Parcourir un arbre de différentes façons (ordres infixe, préfixe ou suffixe ; ordre en largeur d'abord).
6. Rechercher une clé dans un arbre de recherche, insérer une clé.

9 Bases de données - Langage SQL - PJ

1. Identifier les services rendus par un système : persistance des données, gestion des accès concurrents, efficacité de traitement des requêtes, sécurisation des accès.
2. Identifier les composants d'une requête.
3. Construire des requêtes d'interrogation à l'aide de : SELECT, FROM, WHERE, JOIN.
4. Construire des requêtes d'insertion et de mise à jour à l'aide de : UPDATE, INSERT, DELETE.

10 Algorithmique – Graphes - PR

1. Parcourir un graphe en profondeur, en largeur.
2. Repérer la présence d'un cycle dans un graphe.
3. Chercher un chemin dans un graphe.

4. Modélisation à l'aide de classes Python
5. Lien avec les protocoles de routage

11 Algorithmique - PJ

1. Écrire un algorithme utilisant la méthode « diviser pour régner ».
2. Utiliser la programmation dynamique pour écrire un algorithme.
3. Étudier l'algorithme de Boyer-Moore pour la recherche d'un motif dans un texte.

=> BAC BLANC 2 (3h 30 min)

=> TP NOTE (1 heure)

12 Architectures matérielles OS et réseaux - Gestion des processus et des ressources - PR

1. Décrire la création d'un processus,
2. Décrire l'ordonnancement de plusieurs processus par le système.
3. Mettre en évidence le risque de l'interblocage (deadlock).

=> BAC

=> TP NOTE (1 heure)

13 Architectures matérielles OS et réseaux - Sécurisation des communications - PJ

1. Décrire les principes de chiffrement symétrique (clef partagée).
2. Décrire les principes de chiffrement asymétrique (avec clef privée/clef publique).
3. Décrire l'échange d'une clef symétrique en utilisant un protocole asymétrique pour sécuriser une communication HTTPS.

14 Architectures matérielles OS et réseaux - Système sur puce - PR

1. Identifier les principaux composants sur un schéma de circuit
2. Identifier les avantages de leur intégration en termes de vitesse et de consommation.