

VUE SYNOPTIQUE DU COURS

420-SN1-RE PROGRAMMATION EN SCIENCES

OBJECTIF INTÉGRATEUR

Développer des programmes informatiques dans un contexte scientifique

Objectif d'apprentissage 1

Se familiariser avec un langage de programmation (Python)

Nombre de semaines : 3

Objectif d'apprentissage 2

Coder un programme de manière structurée

Nombre de semaines : 6

Objectif d'apprentissage 3

Traiter des données à l'aide d'un programme

Nombre de semaines : 6

Savoirs essentiels

- Environnement de développement : éditeur et extensions
- Analyse du problème : détermination des données d'entrée, des données de sortie et de la nature des traitements
- Algorithmes en pseudo-code ou en ordigramme
- Règle de nommage
- Syntaxe et erreurs de compilation
- Variable et types de données
- Opérateurs : de relation, arithmétiques et logiques
- Expressions et priorité des opérateurs
- Lecture de données et affichage textuel des résultats.
- Vérification du fonctionnement correct du programme

Savoirs essentiels

- Traduction de l'algorithme en code Organisation logique des instructions
- Structures conditionnelles
- Structures répétitives
- Librairies systèmes et fonctions prédéfinies
- Manipulation d'objets de type prédéfini
- Sous-programmes : Déclaration, appel, passage de paramètres et retour
- Portée des variables
- Jeux d'essai simples
- Type d'erreurs
- Débogage et trace d'exécution du programme
- Rigueur dans la programmation et le débogage

Savoirs essentiels

- Traduction de l'algorithme en code Organisation logique des instructions
- Structures conditionnelles
- Structures répétitives
- Librairies systèmes et fonctions prédéfinies
- Manipulation d'objets de type prédéfini
- Sous-programmes : Déclaration, appel, passage de paramètres et retour
- Portée des variables
- Jeux d'essai simples
- Type d'erreurs
- Débogage et trace d'exécution du programme
- Rigueur dans la programmation et le débogage

VUE SYNOPTIQUE DU COURS

Évaluation 1 (10%)	Évaluation 2 (30%)	Évaluation 3 (40%)
<ul style="list-style-type: none">• Mini-test pseudo-code / ordinogramme (semaine 4)	<ul style="list-style-type: none">• Examen en classe (Moodle): python de base, trace d'exécution (semaine 10)	<ul style="list-style-type: none">• Projet intégrateur, fil rouge (semaines 11 à 15)
Évaluation en continu (20%)		
<ul style="list-style-type: none">• Mini-quizz hebdomadaires (10 min)		

INTÉGRATION DES APPRENTISSAGES (40%)

- L'élève développe un projet informatique de traitements et d'affichage de données dans le contexte scientifique (p. ex. simulation physique, représentation graphique, animation) avec les autres disciplines du programme (Biologie, chimie, physique, mathématiques).
- L'élève écrit l'algorithme de son projet.
- Par la suite, il transpose son algorithme en code dans un programme structuré qui lit les données authentiques à partir d'un fichier structuré tout en respectant la syntaxe du langage et en utilisant les traitements nécessaires (tri, groupement, analyse de données).
- Le programme informatique affiche le résultat sous une forme appropriée selon le problème scientifique. L'élève teste et corrige son programme au besoin. Il fait état de sa démarche et de ses résultats dans un rapport en utilisant, par exemple, la programmation lettrée.

CRITÈRES D'ÉVALUATION

- Compétence numérique
- Rigueur scientifique
- Reconnaissance appropriée des concepts impliqués dans un problème
- Décomposition cohérente de l'algorithme
- Utilisation appropriée des types de données de base et des tableaux
- Respect de la syntaxe et des conventions du langage de programmation
- Utilisation appropriée de bibliothèques
- Repérage des erreurs de fonctionnement