

Document d'intention pédagogique - 3h/semaine

UAA11 Programmation procédurale

Les UAA et leur(s) compétence(s)

ANNÉE		INTITULÉ DE L'UAA	COMPÉTENCE(S) VISÉE(S) DANS L'UAA	UAA en prérequis
3 ^e ANNÉE	UAA2	Programmation séquentielle	▪ Programmer une séquence d'instructions pour répondre à un besoin défini.	
4 ^e ANNÉE	UAA5	Programmation impérative	▪ Développer une application non orientée objet sur la base d'un cahier des charges intégrant des chaînes de caractères, des fonctions prédéfinies, des structures alternatives et répétitives.	UAA2
5 ^e ANNÉE	UAA11	Programmation procédurale	▪ Développer une application non orientée objet sur la base d'un cahier des charges intégrant des fonctions personnalisées, des structures imbriquées et un tableau à plusieurs dimensions.	UAA5, UAA2

UAA en prérequis

- ✓ L'UAA11 sera travaillée en classe de 5^{ème}
- ✓ Les UAA2, 5 sont en prérequis de l'UAA11. Pour les élèves non dispensés, elles devront être travaillées à la maison en suivant la progression de l'UAA11.

Critères de réussite

- Les **matières** seront données sous forme de cours, exercices et/ou travaux pratiques ; elles seront évaluées lors de **CS** ou sous forme de projets dont les critères d'évaluations des compétences vous seront précisés au cours de l'année.
- **Cote de réussite: 5 (65%)** sur la moyenne des contrôles sommatifs et des examens (voir plus loin)

Le travail scolaire

Comportement

La réussite de ton année dépend pour beaucoup de ton travail et de ton comportement en classe. Ce dernier doit être basé sur le respect de tous (professeur, élèves) et du matériel.

Tu t'engages donc à respecter les règles suivantes:

- Avant de rentrer en classe, tu te ranges calmement dans le couloir et tu t'installes à ta place dans le calme.
- Tu respectes les différentes phases du cours :
 - Durant les explications théoriques, le silence en classe est total. Si tu dois prendre la parole, tu la demandes en levant la main. Les autres élèves écoutent la question et la réponse.

- Durant les exercices, tu peux parler avec ton voisin à condition que cela concerne le cours et que cela ne perturbe pas la classe.
- Si tu fournis un travail personnel pendant le cours, tu peux déjà bien assimiler la matière. Copier un exercice du tableau ne sert pas à grand-chose, tu dois le faire en même temps que le professeur ou seul si on te laisse le temps.
- Toute intervention sur une machine sera soumise à l'autorisation du professeur (arrêt, redémarrage, installation de logiciel, changement de configuration, téléchargement ...)
- Consulter volontairement des sites se rapportant aux sujets suivants verra l'utilisateur sanctionné (exclusion du cours, ...) : les atteintes aux bonnes mœurs, la publicité pour des offres de services à caractère sexuel; le racisme, la xénophobie et la négation du génocide nazi; la provocation à commettre des crimes et délits; les associations de malfaiteurs; les jeux et paris, les stupéfiants.

Evaluations

La réussite de ton année dépend pour beaucoup de ton travail régulier.

Un conseil : à la fin de chaque journée, revois le cours qui a été abordé dans la journée; cela facilitera ton travail et tes révisions au moment de l'évaluation.

- **Exercices à la maison et devoirs**
Les préparations sont obligatoires ; elles n'interviennent pas dans ton évaluation mais elles montrent ta volonté de travailler et sont indispensables à ton apprentissage.
(Une question de devoir peut de nouveau être posée en CS)
- **Evaluations formatives**
Elles ont lieu en cours d'apprentissage et vise à apprécier les progrès accomplis et à comprendre la nature des difficultés que tu rencontres, en te permettant de tirer des leçons positives des erreurs commises et de les corriger.
- **Les évaluations sommatives ou CS (Contrôle Sommatif) en cours d'année:**
Le CS a toujours lieu à la fin d'une séquence d'apprentissage et pas forcément lors d'une session d'examens. Ce contrôle (écrit ou oral) établit le bilan des compétences acquises.
Il est toujours annoncé.
Une absence à une évaluation sommative entraîne la représentation du contrôle lors de la session d'examens suivante.
- **Sanctions de la triche pendant un CS**

Toute **triche (*)** à un CS entraînera une sanction : **TI au CS** et tu ne pourras plus être évalué sur ce CS pendant les sessions de rattrapage en cours d'année, mais seulement dans le cadre de l'examen de juin. Cette sanction peut être plus lourde : elle peut être disciplinaire avec une communication aux parents et tu peux aussi être interdit de rattrapage à tous tes CS jusqu'à la session de juin.

(*) : Est considéré comme triche : toute découverte d'un stratagème préparé en vue d'une triche, tout non-respect des consignes qui induit un risque de triche (exemple : un smartphone laissé dans une poche de vêtement, ou sur la table, un câble internet branché sur le pc), toute découverte de triche pendant la correction (exemple : une réponse identique à un autre élève ou rédigée par chatgpt)

- **Les examens de synthèse ou projet.**
Aux examens de décembre ou de juin, l'élève sera évalué par un examen de synthèse reprenant l'ensemble de la matière vue.
- En fin d'année, le professeur évalue les compétences de l'élève sur base de ces évaluations sommatives, à savoir :

✓ **50% pour les CS en cours d'année**

✓ **50% pour l'examen de juin.**

- **Remédiation** : voir le règlement de l'établissement.

Si ton attitude est positive et que tu montres ta volonté à travailler, ton professeur s'engage à faire le maximum pour te faire acquérir les compétences de la discipline et te faire réussir ton année.

Signature du professeur

Signature de l'élève

Signature des parents
ou du responsable légal

Contenu, compétences et savoirs à acquérir et à exercer.

Informatique

3^{ème} degré Technique de Transition - cours à 3 périodes/sem. , D/2021/7362/3/10

UAA11 Programmation procédurale

✓ *Compétence*

Développer une application non orientée objet sur la base d'un cahier des charges intégrant des fonctions personnalisées, des structures imbriquées et un tableau à plusieurs dimensions.

✓ *Processus*

APPLIQUER (A)	TRANSFÉRER (T)
<ul style="list-style-type: none"> ▪ Lire un algorithme intégrant des structures imbriquées. ▪ Traduire un algorithme intégrant des structures imbriquées dans un langage de programmation, en respectant la syntaxe. ▪ Programmer en utilisant un tableau à plusieurs dimensions. ▪ Programmer en recourant à des fonctions personnalisées. ▪ Commenter les lignes de codes. ▪ Tester le programme conçu. 	<ul style="list-style-type: none"> ▪ Extraire d'un cahier des charges les informations nécessaires à la programmation. ▪ Écrire un algorithme intégrant des structures imbriquées. ▪ Programmer en recourant aux fonctions personnalisées, instructions et types de données nécessaires au développement d'une application. ▪ Corriger un programme défaillant. ▪ Améliorer un programme pour répondre à un besoin défini.
CONNAITRE (C)	
<ul style="list-style-type: none"> ▪ Expliquer la notion de portée d'une variable. ▪ Différencier un langage interprété d'un langage compilé. ▪ Expliquer la structure d'un tableau. ▪ Expliquer la syntaxe d'utilisation des fonctions principales d'un tableau, dont le calcul de la taille, l'insertion et la suppression d'un élément. ▪ Associer les fonctions principales d'un tableau à leur utilisation. ▪ Expliquer la syntaxe d'utilisation de fonctions personnalisées. 	

UAA2 Programmation séquentielle (prérequis à l'UAA11)

✓ Compétence

Programmer une séquence d'instructions pour répondre à un besoin défini

✓ Processus

APPLIQUER (A)	TRANSFÉRER (T)
<ul style="list-style-type: none"> ▪ Appliquer les règles de syntaxe et les conventions spécifiques à un langage de programmation. ▪ Déclarer une variable en appliquant les règles et les conventions. ▪ Lire un logigramme* d'actions d'un objet réel ou virtuel intégrant structure répétitive et opérateurs logiques. ▪ Traduire un logigramme* dans un langage de programmation textuel*. ▪ Utiliser des fonctions prédéfinies (bibliothèque) en vue d'animer un objet réel ou virtuel. ▪ Tester la séquence d'instructions conçue. ▪ Commenter* des lignes de codes. 	<ul style="list-style-type: none"> ▪ Écrire un logigramme* d'actions d'un objet réel ou virtuel intégrant structure répétitive et opérateurs logiques. ▪ Améliorer une séquence pour répondre à un besoin défini. ▪ Corriger une séquence défectueuse proposée pour atteindre un but défini.
CONNAÎTRE (C)	
<ul style="list-style-type: none"> ▪ Associer les opérateurs de recherche " *, +, -, « », et, ou " à leur fonction ▪ Expliciter en contexte les notions : + <ul style="list-style-type: none"> • Droit à la vie privée, droit à l'oubli, droit de retrait, droit à la propriété, droit d'usage, droit à l'image, liberté d'expression/censure/modération, licence open source • Systèmes d'exploitation, fenêtre, dossier, fichier, extensions, explorateur de fichiers, "en ligne", "hors ligne", logiciel, application, droits d'accès, propriétaire, administrateur, compte utilisateur, • session • Internet, Web, navigateurs Web et leurs fonctions dont signet ou marque-page, favori, moteurs de recherche, barre de recherche, extensions (ou plug in ou add on), site Web, adresse email, historique ▪ Décrire les caractéristiques spécifiques à la taille et au format de fichiers ▪ Identifier des éléments nécessaires au questionnement de la fiabilité d'une source. 	

UAA5 Programmation séquentielle (prérequis à l'UAA11)

✓ Compétence

Développer une application non orientée objet sur la base d'un cahier des charges* intégrant des chaînes de caractères, des fonctions prédéfinies, des structures alternatives et répétitives.

✓ Processus

APPLIQUER (A)	TRANSFÉRER (T)
---------------	----------------

<ul style="list-style-type: none"> ▪ Lire un algorithme* intégrant des structures, alternative et répétitive. ▪ Traduire un algorithme* dans un langage de programmation en respectant sa syntaxe. ▪ Vérifier et valider les données entrantes. ▪ Programmer en utilisant des chaînes de caractères et leurs fonctions prédéfinies. ▪ Programmer en utilisant une structure alternative. ▪ Programmer en utilisant conjointement des structures alternatives et répétitives. ▪ Commenter* des lignes de codes. ▪ Tester le programme conçu. 	<ul style="list-style-type: none"> ▪ Extraire d'un cahier des charges* les informations nécessaires à la programmation. ▪ Écrire un algorithme* intégrant des structures alternatives et répétitives. ▪ Programmer en recourant aux instructions et types de données nécessaires au développement d'une application. ▪ Corriger un programme défaillant. ▪ Améliorer un programme pour répondre à un besoin défini.
CONNAITRE (C)	
<ul style="list-style-type: none"> ▪ Expliquer la notion d'entrée et de sortie. ▪ Expliquer la notion de programmation impérative*. ▪ Expliquer la notion de structure alternative. ▪ Expliquer la syntaxe d'utilisation des fonctions prédéfinies associées à une bibliothèque. ▪ Expliquer la syntaxe d'utilisation des fonctions principales associées à des chaînes de caractères dont la ▪ longueur de chaîne, un caractère à un indice donné. 	