

Pourquoi cette UE ?

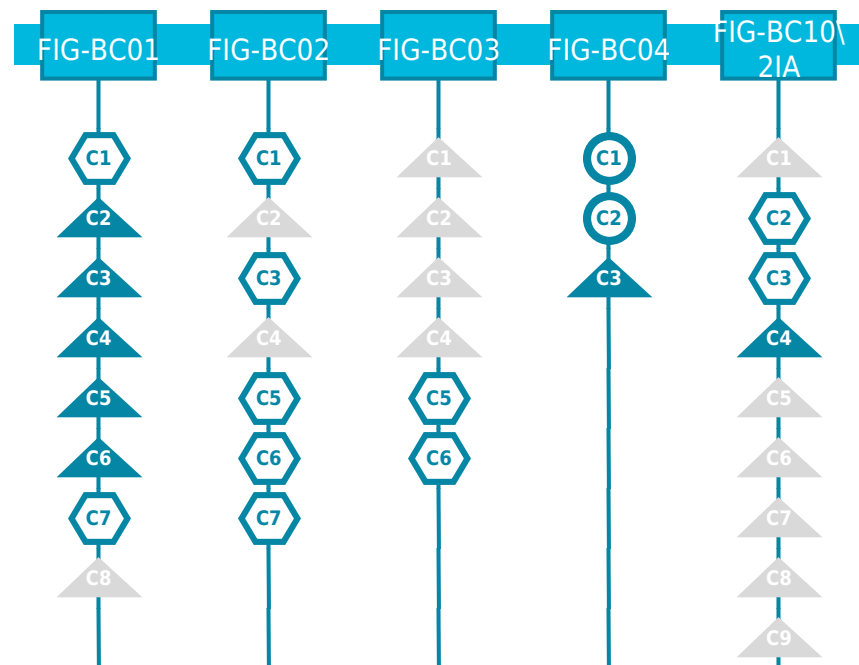
Les traitements informatiques, pour être performants, impliquent que les structures de données employées soient adaptées à l'application visée et à son contexte. La complexité des algorithmes doit être optimisée et les langages de programmation adaptés au contexte d'usage. Des outils peuvent être utilisés pour assurer une bonne qualité du code. Après avoir posé les bases de l'informatique théorique, ce module aborde ces différents aspects. Les applications, dans le cadre de cet enseignement, reposant principalement sur le langage C, une introduction à ce langage est également proposée.

Éléments constitutifs de l'UE

		coefficient
2IA_8_2-1 Introduction à l'informatique théorique		1
2IA_8_2-2 Programmation C		1
2IA_8_2-3 Complexité des algorithmes et récursivité		1
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
50	15	4

Alignement curriculaire

Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?



- BC1 L'UE ne contribue pas à ce bloc de compétences
- BC1 L'UE contribue à ce bloc de compétences
- C1 Compétence non adressée dans cette UE
- C1 Compétence mise en œuvre dans cette UE
- C1 Compétence enseignée dans cette UE
- C1 Compétence évaluée dans cette UE
- C1 Compétence enseignée et évaluée dans cette UE

2IA_8_2 Algorithmique et complexité	FIG
2IA_8_2-1 Introduction à l'informatique théorique	S8

Contexte et enjeux de l'enseignement

Ce cours se destine à des étudiants généralistes ayant des connaissances de base en informatique, auxquels il est nécessaire de montrer que l’informatique s’appuie également sur des fondements théoriques solides et accessibles

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Connaissances mathématiques et informatique de L3

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	9
Cours intégré (cours + TD)	
TD	0
TP	0
Projets	0
Travail en autonomie encadré	
Contrôles et soutenances	1
Travail personnel	3

Objectifs pédagogiques	Activités	Évaluations et retours faits aux élèves
(à la fin de cet enseignement, l'étudiant sera capable de ...)	(CM, TD, TP, projet, sortie terrain, etc.)	(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)
Connaître les limites théoriques aux systèmes informatisés. Savoir s'appuyer sur des modèles théoriques pour prouver certaines propriétés des programmes. Connaître le socle commun aux langages impératifs.	Les enseignements sont proposés sous la forme de cours/TD.	Contrôle écrit individuel 1h Consultation des copies. Délai de correction : 3 semaines.

Plan de cours

- Historique rapide.
- Structures mathématiques discrètes (ensembles, mots, graphes).
- Premiers modèles de calculs : automates finis et machine de Turing. Formalisation du calcul de fonctions. Problèmes de décision.
- Limites du calcul : indécidabilité. Problème de l'arrêt. Thèse de Church.
- Second modèle de calcul : fonctions récursives.
- Grands problèmes ouverts en informatique théorique.

Ressources et références

1 Polycopié

2IA_8_2 Algorithmique et complexité	FIG
2IA_8_2-2 Programmation C	S8

Contexte et enjeux de l'enseignement

Ce langage "bas niveau" reste un outil incontournable lorsque l'on veut maîtriser finement les ressources CPU. Il est suffisamment typé pour représenter les variables de nombreux problèmes réels afin d'implémenter des algorithmes de résolutions efficaces. La notion de pointeurs permet également de réaliser des calculs complexes en mémoire centrale. Cet enseignement mettra aussi en avant l'étude des communications entre périphériques via le port série, un aspect crucial pour le développement de projets embarqués et l'acquisition.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Développement d'algorithmes élémentaires dans un langage procédural

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	10
Cours intégré (cours + TD)	
TD	0
TP	10
Projets	0
Travail en autonomie encadré	0
Contrôles et soutenances	0
Travail personnel	6

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Connaitre le langage C et le développement Arduino et savoir le mettre en application pour résoudre un problème donné.

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Les enseignements sont proposés sous le forme de cours et de TP.

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Évaluation des comptes-rendus de TP réalisés en séance

Mise à disposition des corrections, consultation des copies 3 semaines après la dernière séance

2IA_8_2 Algorithmique et complexité	FIG
2IA_8_2-2 Programmation C	S8

Plan de cours

Compilation séparée, édition des liens, et utilisation de bibliothèques. Implémentation d'un algorithme de calcul et d'une interface graphique. Programmation en environnement Arduino.

Ressources et références

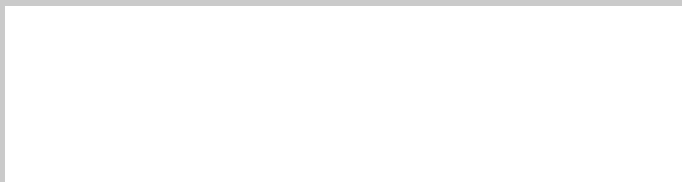
Support de cours et références bibliographiques

2IA_8_2 Algorithmique et complexité	FIG
2IA_8_2-3 Complexité des algorithmes et récursivité	S8

Contexte et enjeux de l'enseignement

Ce cours est un complément à la programmation dans un langage fonctionnel. Il met l'accent sur l'écriture de fonctions récursives tout en sensibilisant le programmeur aux notions de complexité spatiale et temporelle. L'objectif de ce cours est, d'une part, d'apporter une meilleure compréhension d'un "langage à pile" et d'acquérir une première expérience dans l'écriture très expressive de fonctions récursives. D'autre part, ce cours présente les concepts élémentaires à l'évaluation, a priori, de la complexité du code développé.

Prise en compte des dimensions socio-environnementales



Prérequis

Les étudiants doivent avoir une bonne expérience avec un environnement de développement tel que Visual C/C++ et bien connaître les notions de compilation séparée et d'édi

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	10
Cours intégré (cours + TD)	
TD	0
TP	10
Projets	0
Travail en autonomie encadré	0
Contrôles et soutenances	0
Travail personnel	6

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Évaluer le coût en temps et en ressource mémoire d'un algorithme face aux données qu'il est sensé traiter.
Décider, avant codage, si un algorithme est réaliste ou pas.
Exploiter la souplesse et l'efficacité d'un langage à "pile", et exprimer en quelques lignes de code C des traitements relativement complexes.

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Les enseignements sont prévus pour 60 élèves. Les TP seront réalisés sur les ordinateurs personnels de ces derniers.
Le découpage est prévu comme suit :
- 10h de cours
- 10h de TP

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

TP à rendre ; la note finale sera la moyenne des comptes-rendus de TP

3 semaines après la dernière séance.

2IA_8_2 Algorithmique et complexité	FIG
2IA_8_2-3 Complexité des algorithmes et récursivité	S8

Plan de cours

Complexité des algorithmes.
Exploration arborescente.

Ressources et références

Slides du cours et références bibliographiques