

**Pourquoi cette UE ?**

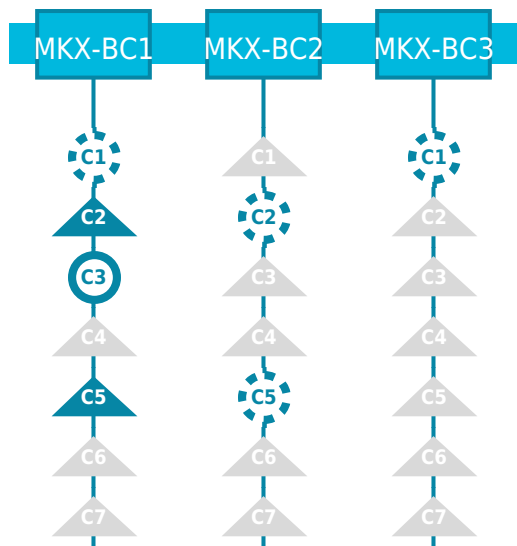
Ce module apporte les bases scientifiques ou technologiques nécessaires pour aborder les enseignements de l'Unité d'Enseignement Génies Electronique-Automatique-Informatique du volet Métier.

**Eléments constitutifs de l'UE**

|                                                          |                                      | coefficient   |
|----------------------------------------------------------|--------------------------------------|---------------|
| MKX_6_2-1 Langage de programmation et modélisation objet |                                      | 1             |
| MKX_6_2-2 Langage de programmation et algorithmique      |                                      | 1             |
| Volume d'heures d'enseignement encadré                   | Volume d'heures de travail personnel | Nombre d'ECTS |
| 62                                                       | 20                                   | 3             |

## 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  |

|                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------|------------|
| <b>MKX_6_2 Bases Scientifiques ou Technologiques : Génies Electronique - Automatique - Informatique</b> | <b>MKX</b> |
| <b>MKX_6_2-1 Langage de programmation et modélisation objet</b>                                         | <b>S6</b>  |

Contexte et enjeux de l'enseignement

Les élèves doivent être capables de faire un lien, au sein d’un système, entre la partie mécanique, la partie électronique (« hardware »), la partie informatique (« software ») et la partie automatique (« commande dynamique »). Les élèves aborderont dans ce module la programmation objet avec le langage Java L'objectif de ce cours est d'apprendre à modéliser une application informatique en objet grâce à UML (diagrammes de classes essentiellement, dans une moindre mesure de cas d'utilisation, de séquences simples, afin de leur donner les compétences clés en programmation objet pour les projets de mécatronique de 2A (Conception du projet fil rouge, UE élective Systèmes embarqués) puis de 3A (Réalisation du projet fil rouge).

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

**Prérequis**

Les bases de la programmation procédurale (Python ou C) sont un plus.

Modalités d'enseignement et d'évaluation

|                              | Nb d'heures |
|------------------------------|-------------|
| Cours                        | 15          |
| Cours intégré (cours + TD)   |             |
| TD                           | 10          |
| TP                           | 2           |
| Projets                      |             |
| Travail en autonomie encadré |             |
| Contrôles et soutenances     | 3           |
| Travail personnel            | 12          |

Objectifs pédagogiques

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

Les élèves doivent être capables de démontrer une compréhension d’algorithmes simples et d'articulation d'un logiciel codé en objet dans un langage de type Java. Ils sauront mobiliser une analyse du besoin puis la conception d'une solution logicielle objet répondant au besoin, au travers d'un diagramme de classes, de diagrammes de cas d'utilisation et de séquence. Ils auront expérimenté comment solliciter l'aide de l'IA générative de façon pertinente et raisonnée.

Activités

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

Alternance de cours magistraux interactifs (16h) et de TD de mise en oeuvre (11h). TP de conception de 2h, par équipes de 3, dont le résultat sera évalué et donnera lieu à une mini-soutenance (2h pour toutes les soutenances).

Les corrections de TD sont fournies après chaque séance (Diagrammes de classes + Code Java).

Évaluations et retours faits aux élèves

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

QCMs formatifs + mises en situation (conception, sollicitation de l'IA générative) et partage de solutions  
3 QCM sommatifs individuels + un rendu de conception en équipe, 2h + soutenance par équipe  
Retours aux élèves après chaque QCM sommatif, retour-échange global en fin d'ECUE.  
En plus des modalités d'évaluation prévues, des évaluations non inscrites à l'emploi du temps pourront avoir lieu. En cas de dysfonctionnement avéré, les évaluations de groupe peuvent être individualisées

## Plan de cours

\* => cours / - => TD / e => évaluation

- \* Présentation de l'enseignant et des élèves (parcours, relation à la programmation informatique)
- \* Un peu d'histoire: d'où vient la programmation objet, pourquoi ? Comparatif des visions procédurale / objet.
- \* Présentation des classes et des objets
- \* Echanges sur des choses qui nous entourent et comment pourrait-on les modéliser en objet
- \* Introduction aux différents types de relation entre objets, illustrés par les exemples vus avant
- \* Niveaux de visibilité des attributs et méthodes - Notion d'encapsulation
- \* Notion de constructeur
- \* Types primitifs et types objet - passage par adresse ou par valeur
- \* Tableaux et listes
- \* Bonnes pratiques de codage en Java
  - TD1 (2h)
- e QCM sommatif 1
- \* Diagrammes de classe de conception - relations entre classes, les cardinalités
- \* Passer du diagrammes de classe de conception à celui d'implémentation - migration d'objets en attributs
- \* Héritage et polymorphisme
  - TD2 (2\*1h30)
- \* Gérer les erreurs grâce aux exceptions / Concevoir les classes d'exception pertinentes pour une application donnée
- \* Classes abstraites et interfaces
- \* Analyse critique de diagrammes de classe proposés par l'IA générative
  - TD3 (2h)
- e QCM sommatif 2
- \* Gestion des flux en Java - Fichiers
- \* Gestion des Threads en Java - principes et mise en œuvre sur des exemples
- \* Introduction aux clients / serveurs et illustration de l'utilité des Threads
  - TD4 (2\*2h)
- \* Diagrammes de cas d'utilisation
- \* Diagrammes de séquence de conception
- e TP de conception noté + soutenance
- e QCM sommatif 3
- \* Echanges sur le cours

## Ressources et références

Les supports pédagogiques sont disponibles en ligne sous Campus.  
Echanges réguliers avec les élèves via Teams.

|                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------|------------|
| <b>MKX_6_2 Bases Scientifiques ou Technologiques : Génies Electronique - Automatique - Informatique</b> | <b>MKX</b> |
| <b>MKX_6_2-2 Langage de programmation et algorithmique</b>                                              | <b>S6</b>  |

## Contexte et enjeux de l'enseignement

Programming concepts are important in mechatronics. Students must be able to connect different parts of a system (mechanical components, hardware controllers, networked systems, etc.) and use them effectively, especially in resource-constrained environments. In this module, students will learn how to program in C and how to effectively manage computer resources. The goal of this course is to learn basic algorithmic concepts, master programming concepts, and know how to define and manipulate data structures.

## Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

### Prérequis

Intermediate Linux/Windows/macOS usage. Basic algorithmization. Computer network basics.

## Modalités d'enseignement et d'évaluation

|                              | Nb d'heures |
|------------------------------|-------------|
| Cours                        | 18          |
| Cours intégré (cours + TD)   |             |
| TD                           | 6           |
| TP                           |             |
| Projets                      | 6           |
| Travail en autonomie encadré |             |
| Contrôles et soutenances     | 2           |
| Travail personnel            | 8           |

## Objectifs pédagogiques

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

Students should be able to handle toolchain for writing, compiling and debugging C code. They should demonstrate understanding of simple algorithms, analyze and write programs in C.

## Activités

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

Combination of lectures (18x 1h) with practical work on discussed topics (6x 1h). Combined with project-based learning in groups of 2 to 4 students (6x 1h) on assigned problems that deepen the understanding of using the principles of the C language.

## Évaluations et retours faits aux élèves

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

Test – individually implemented program in C. In addition to the planned assessment methods, assessments not included in the timetable may also take place. In addition to the planned assessment methods, assessments not included in the timetable may take place. In the event of a proven malfunction, group assessments may be individualized.

## Plan de cours

### Lectures

- Program structure, memory handling, basic variables. Toolchain, basic example program, debugging of code.
- Variable types, type conversions, basic operators, decision making, loops.
- Pointers, functions, scope, arrays, strings.
- Input/output handling, networking.

Day 5: Project finishing and submission, test.

## Ressources et références

Slides and guides/practical tutorials online, program examples. OS virtualbox images.