

# ICOM: Licence 3 Informatique

## Projet : Programmation orientée objet

*Version 0.9 – Bêta*

Année universitaire 2025–2026

### Partie 1 : Contexte et objectif

Dans le cadre de l'évaluation du module **Algorithmique et Programmation Orientée Objet**, vous êtes chargés de concevoir une application exploitant les concepts de la POO vus en CM et en TD. L'objet de votre application et ses fonctionnalités sont libres, mais celle-ci doit respecter au moins les exigences suivantes (Des points bonus pourront être accordés pour des fonctionnalités supplémentaires).

### Partie 2 : Cahier de charge et exigences minimales

1. Implémentation d'au moins **4 classes** en plus d'un programme **main** contenant la création des objets et l'appel des méthodes des classes.
2. Les classes doivent comporter des attributs et des méthodes avec des spécificateurs d'accès **public**, **privé** et **protégé**, justifiés.
3. Pour chacune des classes, fournir un fichier entête (**.hpp**) déclarant les attributs et les prototypes, et un fichier source (**.cpp**) avec l'implémentation.
4. Dans le **main**, l'initialisation et la récupération des valeurs des attributs doivent se faire via **getters** et **setters**.
5. L'application doit contenir des comparaisons entre objets à l'aide des opérateurs (**operator==()**, **operator>()**, **operator<()**, etc.).
6. Au moins une relation d'**association**, d'**agrégation** ou de **composition** entre les classes, ainsi qu'un **héritage**.
7. Présence d'au moins une **méthode virtuelle** (polymorphisme).
8. Gestion de la lecture/écriture depuis/sur des fichiers.
9. Utilisation de **pointeurs**, méthodes **const**, transmissions par **valeur** et par **référence**, avec justification.

### Partie 3 : Étapes et livrables

#### 3.1 Organisation du binôme

Former un binôme. Le projet et les livrables doivent indiquer clairement les deux membres du binôme.

### 3.2 Livrable 1

- Un fichier PDF comprenant un synopsis de l'application : noms des membres, intitulé du projet et une brève description des fonctionnalités prévues.
- Date limite : **Jeudi 4 décembre 2025 à 23h59**.
- Nom du fichier : `Livrable1_NOM1_NOM2.pdf`.
- Dépôt dans l'espace Moodle dédié au Livrable 1 (un seul dépôt par binôme).

### 3.3 Livrable 2

- Une archive `zip` contenant deux dossiers :
  1. Un dossier **rapport** avec un document PDF détaillant les classes (rôles, relations, sémantiques) et un diagramme de classes.
  2. Un dossier **code** avec les fichiers sources : `.cpp`, `.hpp`, le point d'entrée `main.cpp` et l'exécutable `main.exe`.
- Nom de l'archive : `Livrable2_NOM1_NOM2.zip`.
- Date limite : **Mardi 23 décembre 2025 à 23h59**.
- Dépôt dans l'espace Moodle dédié au Livrable 2 (un seul dépôt par binôme).

## Partie 4 : Exemples de projets (Liste non exhaustive)

Voici quelques exemples de projets que nous vous proposons, sachant que la liste n'est pas exhaustive et que vous pouvez ajouter une dimension création numérique aux projets de la catégorie sciences de données et vice versa . À vous de choisir selon le thème qui vous plaît et le projet que vous souhaitez mettre en avant dans vos CV, et libre à vous d'imaginer d'autres projets.

### 4.1 Projets en lien avec le parcours Sciences de données

**Analyse de Séquences Biologiques** : Une application permettant de gérer et analyser des séquences d'ADN ou d'ARN comprenant des classes pour les séquences, gènes, fichiers biologiques, analyseurs, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'import/export de séquences, la recherche de motifs, le calcul de fréquences, la comparaison de séquences, etc. (fonctionnalités données à titre d'exemple).

**Système de Recommandation Simplifié** : Une application de recommandation de films, livres ou produits comprenant des classes pour les utilisateurs, items, notes, moteur de recommandation, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'ajout d'utilisateurs/items, la notation, le calcul de similarités, la génération de recommandations, etc. (fonctionnalités données à titre d'exemple).

**Simulation de Capteurs et Collecte de Données** : Une application simulant un réseau de capteurs comprenant des classes pour les capteurs, le réseau, les données, analyseur, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la lecture/écriture de données, le calcul de statistiques, la détection d'anomalies, la génération de rapports, etc. (fonctionnalités données à titre d'exemple).

**Social Media Trends Tracker** : Une application de suivi des tendances sur les réseaux sociaux comprenant des classes pour les utilisateurs, publications, hashtags, analyseur de tendances, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la création de publications, l'extraction de hashtags, le calcul de popularité, la visualisation des tendances, etc. (fonctionnalités données à titre d'exemple).

**Gestionnaire de Journaux de Sécurité (Security Log Manager)** : Une application de gestion et d'analyse des journaux de sécurité comprenant des classes pour les logs, événements, serveurs, analyseur,

etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'ajout d'événements, le filtrage par type/date, la détection d'anomalies ou d'intrusions, etc. (fonctionnalités données à titre d'exemple).

**Sports Stats Tracker** : Une application d'analyse de performances sportives comprenant des classes pour les joueurs, équipes, matchs, statistiques, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'enregistrement des scores, le calcul de moyennes, la comparaison de joueurs/équipes, la génération de classements, etc. (fonctionnalités données à titre d'exemple).

## 4.2 Projets en lien avec le parcours Création numérique

**NanoLab Intelligence** : Une application de gestion de nanorobots et missions comprenant des classes pour les nanorobots, missions, capteurs, laboratoire, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la création de nanorobots, l'assignation de missions, le suivi des positions, la détection de collisions, l'enregistrement des résultats, etc. (fonctionnalités données à titre d'exemple).

**Space Object Tracker** : Une application de suivi des planètes, astéroïdes et satellites comprenant des classes pour les objets spatiaux, orbites, capteurs, observatoire, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que le calcul de trajectoires, la détection de collisions, le stockage de données, la visualisation des positions, etc. (fonctionnalités données à titre d'exemple).

**Smart City Simulator** : Une application de simulation de ville intelligente comprenant des classes pour les bâtiments, capteurs, routes, gestionnaire d'énergie, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la simulation de consommation, le calcul de flux de trafic, la génération d'alertes, la sauvegarde de l'état de la ville, etc. (fonctionnalités données à titre d'exemple).

**Art Generator Tracker** : Une application d'analyse de créations numériques comprenant des classes pour les œuvres, artistes, galeries, analyseur de styles, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la création d'œuvres, la classification de styles, le suivi de popularité, l'export de rapports, etc. (fonctionnalités données à titre d'exemple).

**Virtual Zoo** : Une application de gestion et suivi d'animaux et habitats comprenant des classes pour les animaux, habitats, soigneurs, visiteurs, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'assignation d'animaux aux habitats, le suivi de la santé, l'enregistrement des interactions, la visualisation de statistiques, etc. (fonctionnalités données à titre d'exemple).

**Système de Suivi des Écosystèmes (EcoTracker)** : Une application de gestion et suivi d'écosystèmes naturels comprenant des classes pour les espèces, habitats, mesures environnementales, observateurs, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'ajout d'espèces, l'enregistrement de mesures, l'analyse des interactions, la sauvegarde des données, etc. (fonctionnalités données à titre d'exemple).

## 4.3 Applications classiques POO

**Gestionnaire de Bibliothèque** : Une application de gestion de bibliothèque comprenant des classes pour les livres, les membres, les emprunts, bibliothécaire, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'ajout de livres, l'emprunt/le retour, la gestion des membres, etc. (fonctionnalités données à titre d'exemple).

**Simulation de Banque** : Une application de simulation bancaire comprenant des classes pour les comptes, clients, transactions, banque, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que les dépôts, retraits, transferts, calculs d'intérêts, historisation des transactions, etc. (fonctionnalités données à titre d'exemple).

**Jeu de Stratégie** : Une application de jeu stratégique comprenant des classes pour les unités, bâtiments, ressources, joueurs, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la construction de bâtiments, l'entraînement d'unités, les combats, la gestion des ressources, etc. (fonctionnalités données à titre d'exemple).

**Système de Gestion d'Événements** : Une application de gestion d'événements comprenant des classes pour les événements, participants, inscriptions, salles, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la création d'événements, la gestion des inscriptions, le contrôle des disponibilités, la génération de listes de participants, etc. (fonctionnalités données à titre d'exemple).

**Application de Gestion de Magasin** : Une application de gestion de magasin comprenant des classes pour les produits, clients, commandes, stock, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que l'ajout de produits, la gestion des commandes, la mise à jour des stocks, le calcul des ventes, etc. (fonctionnalités données à titre d'exemple).

**Système de Réservation de Transport** : Une application de réservation de transports comprenant des classes pour les véhicules, passagers, billets, trajets, etc. (classes données à titre d'exemple), avec des fonctionnalités telles que la création de trajets, la réservation de billets, le calcul des tarifs, la visualisation du planning, l'historisation des réservations, etc. (fonctionnalités données à titre d'exemple).

## Partie 5 : Utilisation des outils d'IA générative

Les étudiants sont autorisés à utiliser des outils d'IA (ChatGPT, Copilot, etc.) mais selon les règles suivantes :

### Brainstorming et conception

Vous pouvez utiliser les outils pour générer des idées, visualiser des concepts ou explorer des architectures possibles. Vous devez **apporter votre réflexion personnelle** : expliquer pourquoi vous avez choisi une solution plutôt qu'une autre et comment vous l'avez adaptée à votre projet.

### Écriture du code

Vous pouvez vous inspirer d'exemples ou demander des suggestions de code à l'IA. **Toute ligne de code utilisée doit être comprise et adaptée par vous-même.** Il est interdit de copier-coller du code sans comprendre son fonctionnement et sans pouvoir l'expliquer.

### Analyse critique

Dans votre rapport, **incluez une section "Utilisation des outils d'IA"** détaillant : Quels outils ont été utilisés et à quelle étape (brainstorming, code, documentation, etc.) ? Comment l'outil vous a aidé ? Les adaptations ou corrections apportées pour que le code fonctionne correctement. Votre réflexion sur la fiabilité ou les limites de l'outil utilisé.

### Responsabilité académique

L'objectif est de développer votre compréhension et vos compétences en programmation orientée objet, pas de déléguer entièrement le travail à l'IA. La bonne pratique est d'utiliser ces outils comme support et non comme solution finale.