

## PLAN DE COURS

**Titre du cours:** Programmation orientée objet en C++

**Cote du cours:** 025908

**Durée du cours:** 42 heures

**Programme:** Technologie du génie informatique

### Description

À la fin de ce cours, l'étudiant sera en mesure d'analyser, modéliser et développer une application simple en utilisant le langage de programmation C++ ainsi que les principes de la programmation orientée objet. De plus, l'étudiant saura employer les subtilités du langage C++ et en comprendra les complexités.

### Pondération

Unité d'apprentissage	Pondération (%)
025951 - Fondements du langage C++	35 %
027230 - Programmation orientée objet, surcharge des opérateurs et gestion de mémoire en C++	35 %
027231 - Héritage, abstraction et polymorphisme en C++	30 %
<b>TOTAL:</b>	<b>100 %</b>

### Système de notation

La note de passage de ce cours est de 60%.

A+ 90-100	B+ 77-79	C+ 67-69	DR 55-59
A 85-89	B 74-76	C 64-66	EC 0-54
A- 80-84	B- 70-73	C- 60-63	

# Division en unités d'apprentissage

## Unité d'apprentissage

**Titre du cours:** Fondements du langage C++

**Cote du cours:** 025951

**Durée du cours:** 14 heures

**Programme:** Technologie du génie informatique

## Description

À la fin de cette unité d'apprentissage, l'étudiant sera en mesure de créer des programmes simples C++ en utilisant des structures de contrôles simples, des fonctions, des tableaux ainsi que des pointeurs. De plus, l'étudiant se familiarisera avec certains logiciels de développement qui facilitent la programmation avec le langage C++.

## Compétences du programme ciblées par ce cours

- Participer à l'analyse, à la conception et au développement des systèmes informatiques, les implanter et en assurer la maintenance;

## Indicateurs de développement

À la fin de ce cours, l'étudiant qui réussit sera en mesure de:

- Développer des applications simples en utilisant le langage de programmation C++.

## Évaluations

### Évaluations formatives

- **Mises en situation**

Durant cette unité d'apprentissage l'étudiant sera confronté à des mises en situation imaginées par le professeur. Dans le but d'identifier et de résoudre les problèmes rencontrés, il devra mettre en pratique les notions présentées, comme l'écriture de programmes simples en C++. Les étudiants pourront valider leurs solutions entre eux ou en les comparant aux solutions du professeur.

- **Études de cas**

Durant cette unité d'apprentissage, l'étudiant sera confronté à des études de cas existantes ou imaginées par le professeur. Il devra analyser du code source en C++ dans le but de le documenter. Pour ce faire, il devra également suivre les bonnes pratiques et conventions de programmation. Les étudiants pourront valider leurs documentations en groupe.

### Évaluation(s) sommative(s)

- **Évaluation pratique**

À la fin de cette unité d'apprentissage, l'étudiant devra faire une évaluation pratique dans laquelle il devra trouver des solutions à différentes mises en situation imaginées par le professeur. L'étudiant devra créer des programmes simples à l'aide du langage C++ pour solutionner ces mises en situations. Cette évaluation se fera de façon individuelle et sera corrigée par le professeur.

### Système de notation

La note de passage de ce cours est de 60%.

A+ 90-100	B+ 77-79	C+ 67-69	DR 55-59
A 85-89	B 74-76	C 64-66	EC 0-54
A- 80-84	B- 70-73	C- 60-63	

## Contenu

- Utilisation d'un logiciel d'environnement de développement intégré pour créer et développer un espace de travail en C++
- Syntaxe de base du langage C++ et de ses particularités par rapport aux autres langages de programmation
- Utilisation des structures de contrôle et des fonctions dans un programme C++
- Utilisation des tableaux et des pointeurs dans un programme C++
- Ressources en ligne facilitant le développement avec le langage C++
- Conventions d'écriture de code de base en C++

## Modalités pédagogiques

- Démonstrations pratiques
- Mises en situation
- Études de cas

## Séquence pédagogique

Modules	Lectures, travaux et évaluations
Environnement de développement Entrées et sorties de la console Variables et portée Structures de contrôle Génération de nombres aléatoires	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative)
Fonctions mathématiques Création de ses propres fonctions Arguments constants et par défaut Fonctions en ligne Surcharge de fonctions Passage d'arguments par référence ou par valeur	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative)
Tableaux simples Tableaux multidimensionnels Énumérations Vecteurs et listes Fonctions pour les vecteurs et listes	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative)
Tableaux de caractères Chaîne de caractères Fonctions pour les chaînes de caractères Algorithmes pour les tableaux Fonctions pour les tableaux de caractères	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative)
Variable de type pointeurs Opérateurs pour les pointeurs Arithmétique des pointeurs Relation entre les pointeurs et les tableaux Convention d'écriture du code	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Étude de cas (évaluation formative)
Évaluation sommative	Évaluation pratique (évaluation sommative)

## Exigences

### Exigences obligatoires

- L'étudiant est responsable de prendre connaissance de son Guide de programme disponible dans le Portail étudiant. Il y trouvera notamment les directives de l'étudiant fournissant des informations précises en ce qui a trait à son cheminement scolaire, l'intégrité scolaire, les règles de conduite, les équivalences et reconnaissances des acquis, l'ajout et l'abandon de cours, le transfert de programme et les évaluations de reprise en cas d'échec.
- L'étudiant a la responsabilité d'être présent au cours, de participer et d'apporter tout outil, document ou fourniture exigé.
- L'étudiant dont le comportement est jugé inapproprié peut se voir refuser l'accès au cours.
- L'intégrité scolaire fait partie intégrante de l'apprentissage de l'étudiant. Elle vise à promouvoir un environnement où l'honnêteté intellectuelle

occupe une place de choix. Le plagiat et la fraude scolaire constituent un manquement à l'intégrité scolaire. Toute forme de plagiat et de fraude scolaire est interdite et sera sanctionnée selon les dispositions indiquées dans les Directives pédagogiques du Collège.

- Tous les devoirs et les travaux doivent être remis à la date indiquée.
- La présentation des devoirs et des travaux doit se conformer aux exigences indiquées dans le cours.

#### **Exigences particulières**

- SO

## **Cohérence pédagogique**

#### **Résultats d'apprentissage de la formation professionnelle (RAFP)**

Ce cours contribue à l'atteinte des connaissances, aptitudes et attitudes suivantes :

- Identifier, analyser, concevoir, développer, mettre en oeuvre, vérifier et documenter les exigences reliées au contexte de l'informatique.
- Diagnostiquer, dépanner, document et surveiller les problèmes techniques.
- Explorer les tendances émergentes pour relever les défis techniques.

#### **Résultats d'apprentissage relatifs à l'employabilité (RARE)**

Ce cours contribue à l'atteinte des connaissances, habiletés et attitudes suivantes qui sont essentielles à la réussite professionnelle ainsi qu'à l'apprentissage continu :

- Appliquer une approche systématique de résolution de problèmes (no.5).
- Utiliser une variété de stratégies pour prévoir et résoudre des problèmes (no.6).

## Unité d'apprentissage

**Titre du cours:** Programmation orientée objet, surcharge des opérateurs et gestion de mémoire en C++

**Cote du cours:** 027230

**Durée du cours:** 14 heures

**Programme:** Technologie du génie informatique

## Description

À la fin de cette unité d'apprentissage, l'étudiant sera en mesure de structurer et organiser ses projets sous la forme de classes et d'objets. De plus, il sera apte à planifier la structure de son développement à l'aide des diagrammes de classe et de gérer adéquatement la mémoire requise par ses programmes. Finalement, l'étudiant pourra utiliser la surcharge des opérateurs pour faciliter l'utilisation de ses classes et objets.

## Compétences du programme ciblées par ce cours

- Participer à l'analyse, à la conception et au développement des systèmes informatiques, les implanter et en assurer la maintenance;

## Indicateurs de développement

À la fin de ce cours, l'étudiant qui réussit sera en mesure de:

- Utiliser le langage de programmation C++ pour organiser le code sous la forme d'objets et employer la surcharge des opérateurs lorsque nécessaire.

## Évaluations

### Évaluations formatives

- **Mises en situations**

Durant cette unité d'apprentissage l'étudiant sera confronté à des mises en situation imaginées par le professeur. Dans le but d'identifier et de résoudre les problèmes rencontrés, il devra mettre en pratique les notions présentées, comme l'écriture de programmes organisés sous forme de classe et objets en C++, la surcharge des opérateurs ou encore la bonne gestion de mémoire. Les étudiants pourront valider leurs solutions entre eux ou en les comparant aux solutions du professeur.

- **Études de cas**

Durant cette unité d'apprentissage, l'étudiant sera confronté à des études de cas existantes ou imaginées par le professeur. Il devra analyser du code source en C++ ou des diagrammes de classes pour mieux documenter et comprendre les concepts de la programmation orientée objet ainsi que de la gestion de mémoire. Pour ce faire, il devra suivre les bonnes pratiques et conventions de programmation ainsi que de gestion de mémoires. Les étudiants pourront valider et comparer leurs analyses entre eux ou en groupe.

### Évaluation(s) sommative(s)

- **Projet**

Durant cette unité d'apprentissage l'étudiant devra créer un projet C++ qui sera planifié et organisé sous la forme de classes et objets. De plus, l'étudiant devra utiliser la surcharge des opérateurs et bien gérer la mémoire de son programme. L'étudiant sera évalué par le professeur à chaque semaine sur sa progression dans le projet.

### Système de notation

La note de passage de ce cours est de 60%.

A+ 90-100

B+ 77-79

C+ 67-69

DR 55-59

A 85-89

B 74-76

C 64-66

EC 0-54

A- 80-84

B- 70-73

C- 60-63

## Contenu

- Planification de la structure d'un programme à l'aide des diagrammes de classe
- Syntaxe pour créer des classes et des structures ainsi qu'instancier des objets en C++
- Terminologie reliée à la programmation orientée objet
- Utilisation de l'encapsulation et des conventions d'écriture de classes en C++
- Création et organisation de programmes complexes organisés sous la forme de classes
- Gérer la mémoire d'un programme C++ dans le but d'éviter les fuites

## Modalités pédagogiques

- Démonstrations pratiques
- Mises en situation
- Études de cas
- Projet

## Séquence pédagogique

Modules	Lectures, travaux et évaluations
Fondement de la programmation orientée objet Diagramme de classe Création de structure de données et de classes Méthodes	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Étude de cas (évaluation formative) Travail sur le projet
Encapsulation Constructeur Accesseur et mutateur Pointeur "this"	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Travail sur le projet
Allocation dynamique de mémoire Destructeur Arguments et objets constant	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Étude de cas (évaluation formative) Travail sur le projet
Type de surcharge d'opérateurs Surcharge des opérateurs de flux Surcharge des opérateurs d'affectation Surcharge des autres opérateurs Ressources en lignes pour la surcharge	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Travail sur le projet
Évaluation sommative	Présentation du projet

## Exigences

### Exigences obligatoires

- L'étudiant est responsable de prendre connaissance de son Guide de programme disponible dans le Portail étudiant. Il y trouvera notamment les directives de l'étudiant fournissant des informations précises en ce qui a trait à son cheminement scolaire, l'intégrité scolaire, les règles de conduite, les équivalences et reconnaissances des acquis, l'ajout et l'abandon de cours, le transfert de programme et les évaluations de reprise en cas d'échec.
- L'étudiant a la responsabilité d'être présent au cours, de participer et d'apporter tout outil, document ou fourniture exigé.
- L'étudiant dont le comportement est jugé inapproprié peut se voir refuser l'accès au cours.
- L'intégrité scolaire fait partie intégrante de l'apprentissage de l'étudiant. Elle vise à promouvoir un environnement où l'honnêteté intellectuelle occupe une place de choix. Le plagiat et la fraude scolaire constituent un manquement à l'intégrité scolaire. Toute forme de plagiat et de fraude scolaire est interdite et sera sanctionnée selon les dispositions indiquées dans les Directives pédagogiques du Collège.
- Tous les devoirs et les travaux doivent être remis à la date indiquée.
- La présentation des devoirs et des travaux doit se conformer aux exigences indiquées dans le cours.

### Exigences particulières

- SO

## Cohérence pédagogique

### Résultats d'apprentissage de la formation professionnelle (RAFP)

Ce cours contribue à l'atteinte des connaissances, aptitudes et attitudes suivantes :

- Identifier, analyser, concevoir, développer, mettre en oeuvre, vérifier et documenter les exigences reliées au contexte de l'informatique.
- Diagnostiquer, dépanner, document et surveiller les problèmes techniques.
- Explorer les tendances émergentes pour relever les défis techniques.

### Résultats d'apprentissage relatifs à l'employabilité (RARE)

Ce cours contribue à l'atteinte des connaissances, habiletés et attitudes suivantes qui sont essentielles à la réussite professionnelle ainsi qu'à l'apprentissage continu :

- Appliquer une approche systématique de résolution de problèmes (no.5).
- Utiliser une variété de stratégies pour prévoir et résoudre des problèmes (no.6).

## Unité d'apprentissage

**Titre du cours:** Héritage, abstraction et polymorphisme en C++

**Cote du cours:** 027231

**Durée du cours:** 14 heures

**Programme:** Technologie du génie informatique

## Description

À la fin de cette unité d'apprentissage, l'étudiant sera en mesure d'utiliser l'héritage et le polymorphisme en C++ pour favoriser la réutilisation de code et la maintenance de projets.

## Compétences du programme ciblées par ce cours

- Participer à l'analyse, à la conception et au développement des systèmes informatiques, les implanter et en assurer la maintenance;

## Indicateurs de développement

À la fin de ce cours, l'étudiant qui réussit sera en mesure de:

- Comprendre l'héritage, ses concepts dérivés ainsi que ses particularités dans la programmation orientée objet afin de développer des applications simples à l'aide du langage de programmation C++.

## Évaluations

### Évaluations formatives

- **Mise en situation**  
Durant cette unité d'apprentissage l'étudiant sera confronté à des mises en situation imaginées par le professeur. Dans le but d'identifier et de résoudre les problèmes rencontrés, il devra mettre en pratique les notions présentées, comme l'utilisation de l'héritage et du polymorphisme en C++. Les étudiants pourront valider leurs solutions entre eux ou en les comparant aux solutions du professeur.
- **Étude de cas**  
Durant cette unité d'apprentissage, l'étudiant sera confronté à des études de cas existantes ou imaginées par le professeur. Il devra analyser du code source en C++ ou des diagrammes de classe pour mieux documenter et comprendre les concepts d'héritage, d'abstraction et de polymorphisme. Pour ce faire, il devra également suivre les bonnes pratiques et conventions de programmation. Les étudiants pourront valider leurs documentations en groupe.

### Évaluation(s) sommative(s)

- **Évaluation pratique**  
À la fin de cette unité d'apprentissage, l'étudiant devra faire une évaluation pratique dans laquelle il devra trouver des solutions à des mises en situation imaginées par le professeur en créant ou complétant des programmes utilisant les concepts de l'héritage et du polymorphisme en C++. Cette évaluation se fera de façon individuelle et sera corrigée par le professeur.

### Système de notation

La note de passage de ce cours est de 60%.

A+ 90-100	B+ 77-79	C+ 67-69	DR 55-59
A 85-89	B 74-76	C 64-66	EC 0-54
A- 80-84	B- 70-73	C- 60-63	

## Contenu

- Syntaxe du code C++ lié à l'héritage et au polymorphisme



- Utilisation de l'héritage pour favoriser la réutilisation de code
- Utilisation du polymorphisme pour favoriser la maintenance du code

## Modalités pédagogiques

- Démonstrations pratiques
- Mises en situation
- Études de cas

## Séquence pédagogique

Modules	Lectures, travaux et évaluations
Concept de l'héritage Héritage dans les diagrammes de classe Héritage dans les classes en C++ Encapsulation protégé	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Étude de cas (évaluation formative)
Transtypage de classes Liaison statique et dynamique Polymorphisme et fonctions virtuelles Classes abstraites Destructeur virtuel	Lecture des notes de cours du professeur Visionnement des capsules vidéo du professeur Mise en situation (évaluation formative) Étude de cas (évaluation formative)
Évaluation sommative	Examen théorique

## Exigences

### Exigences obligatoires

- L'étudiant est responsable de prendre connaissance de son Guide de programme disponible dans le Portail étudiant. Il y trouvera notamment les directives de l'étudiant fournissant des informations précises en ce qui a trait à son cheminement scolaire, l'intégrité scolaire, les règles de conduite, les équivalences et reconnaissances des acquis, l'ajout et l'abandon de cours, le transfert de programme et les évaluations de reprise en cas d'échec.
- L'étudiant a la responsabilité d'être présent au cours, de participer et d'apporter tout outil, document ou fourniture exigé.
- L'étudiant dont le comportement est jugé inapproprié peut se voir refuser l'accès au cours.
- L'intégrité scolaire fait partie intégrante de l'apprentissage de l'étudiant. Elle vise à promouvoir un environnement où l'honnêteté intellectuelle occupe une place de choix. Le plagiat et la fraude scolaire constituent un manquement à l'intégrité scolaire. Toute forme de plagiat et de fraude scolaire est interdite et sera sanctionnée selon les dispositions indiquées dans les Directives pédagogiques du Collège.
- Tous les devoirs et les travaux doivent être remis à la date indiquée.
- La présentation des devoirs et des travaux doit se conformer aux exigences indiquées dans le cours.

### Exigences particulières

- SO

## Cohérence pédagogique

### Résultats d'apprentissage de la formation professionnelle (RAFP)

Ce cours contribue à l'atteinte des connaissances, aptitudes et attitudes suivantes :

- Identifier, analyser, concevoir, développer, mettre en oeuvre, vérifier et documenter les exigences reliées au contexte de l'informatique.
- Diagnostiquer, dépanner, document et surveiller les problèmes techniques.
- Explorer les tendances émergentes pour relever les défis techniques.

### Résultats d'apprentissage relatifs à l'employabilité (RARE)

Ce cours contribue à l'atteinte des connaissances, habiletés et attitudes suivantes qui sont essentielles à la réussite professionnelle ainsi qu'à

l'apprentissage continu :

- Appliquer une approche systématique de résolution de problèmes (no.5).
- Utiliser une variété de stratégies pour prévoir et résoudre des problèmes (no.6).

## Service d'appui et d'adaptation

Les étudiants qui ont ou qui croient avoir une ou des limitations fonctionnelles permanentes ou temporaires qui nécessitent des mesures d'accommodement académiques sont priés de s'inscrire le plus tôt possible auprès du Service La Boussole, situé au local C1030. Un document provenant d'un professionnel de la santé réglementé décrivant la nature et la portée des limitations fonctionnelles est requis pour l'obtention des mesures d'accommodement académiques, mais un étudiant pourrait recevoir des mesures d'accommodement intérimaires dans l'attente de cette documentation. Tous les documents provenant des professionnels de la santé réglementés seront conservés de façon confidentielle à La Boussole et la copie originale sera rendue. Pour plus d'information, veuillez contacter le (613) 742-2483 poste 2090 ou par courriel à [laboussole@collegelacite.ca](mailto:laboussole@collegelacite.ca)

## Reconnaissance des acquis non scolaires

Les étudiants intéressés à se prévaloir du processus de reconnaissance des acquis peuvent obtenir des renseignements relatifs au processus d'évaluation en consultant le Guide de votre programme.