

# **ATTESTATION D'ETUDES COLLÉGIALES**

## **Conception et programmation de sites Web (NWE.0F - 2020)**

**Plan de cours  
Groupe 20632**

**Programmation Web avancée  
582-31B-MA**

**75 heures**

**(2-3-3)**

**Session hiver 2022**

**Professeur :  
Jouhannet, Charles**

[cjouhannet@cmaisonneuve.qc.ca](mailto:cjouhannet@cmaisonneuve.qc.ca)

**COLLÈGE DE MAISONNEUVE  
3800 rue Sherbrooke Est  
Montréal, Québec, H1X 2A2  
Téléphone : 514 254-7131**



**Collège de  
Maisonneuve**

# 1. Présentation du cours

## 1.1 Place du cours dans la formation de l'étudiant

Le cours *Programmation Web avancée* (582-31B) fait partie du champ de compétence Programmation et touche principalement à la programmation côté serveur. Dans ce cours, l'étudiant poursuit ses apprentissages du langage de programmation PHP, des principes de la programmation orientée objet et des types d'architecture de développement Web. Le cours fait la transition entre la programmation procédurale étudiée au cours de Programmation Web dynamiques vers la programmation orientée-objet.

Situé à la 3<sup>e</sup> étape, ce cours a comme préalable le cours *Programmation Web dynamique* (582-21B). Il est préalable au cours *Projet web 1* (582-32W).

Ce cours vise la poursuite des apprentissages en programmation côté-serveur, principalement ce qui touche la programmation orientée objet, l'architecture de développement et les patrons de conception. Le cours explorera principalement les concepts de classes, d'objets, d'héritage, de polymorphisme, d'encapsulation, d'instanciation de classe, l'utilisation de fonctions prédéfinies et de bibliothèques de code à partir d'un langage de programmation orientée-objet côté serveur. Ces concepts seront appliqués aux requêtes HTTP afin d'interagir avec une base de données relationnelle.

À la fin du cours, l'étudiant sera en mesure de créer un site web dynamique, interagissant avec une base de données, en appliquant un patron de conception orientée objet.

Le cours Programmation Web avancée fait partie de la compétence Programmation et est situé à la 3<sup>e</sup> étape du parcours de l'étudiant, juste avant la réalisation du premier projet. Il finalise la compétence Effectuer la programmation d'un site Web (EHPA) et approfondit la compétence Effectuer le développement de la partie serveur d'un site Web (EHP2). À ce titre, l'autonomie de l'étudiant sera mobilisée grandement. À la fin du cours, il devra être en mesure d'effectuer efficacement une recherche dans la documentation et trouver des solutions aux problèmes rencontrés. La rigueur, la logique et la créativité seront toutefois toujours mise de l'avant.

## 1.2 Les compétences à développer

Énoncé de la compétence	Éléments de compétence associés au cours	Heures
Effectuer la programmation d'un site Web (EHPA)	1. Analyser la demande. 2. Programmer les éléments du site Web. 3. Résoudre les erreurs de programmation. 4. Vérifier la qualité de la programmation. 5. Documenter la programmation. 6. Gérer les versions de code source	15
Effectuer le développement de la partie serveur d'un site Web (EHP2)	1. Traiter les requêtes d'un navigateur Web. 2. Interagir avec une base de données en-ligne. 3. Répondre aux requêtes d'un navigateur Web.	60

## 2. Démarche d'enseignement et d'apprentissage

---

Les méthodes pédagogiques privilégiées à l'intérieur de ce cours seront les suivantes :

- Exposés magistraux appuyés de renforts visuels et de notes de cours sous forme de vidéo, de PDF ou d'autres références;
- Exercices pratiques dirigés ou de résolution de problèmes en classe;
- Exercices à faire à la maison, dont certains seront évalués, afin d'inciter l'étudiant à mettre en pratique régulièrement les notions acquises;

Des exercices d'évaluation formative sont aussi prévus tout au long de la session dont :

- Retours sur des exercices pratiques, des examens ou autres;
- Suivi des étudiants. L'étudiant doit réaliser des travaux pratiques au cours de la session dans le cadre des périodes de production des travaux pratiques. Lors de ces moments, il pourra recevoir des appréciations et des remarques sur son travail effectué et sur la qualité de ses apprentissages.

### 3. Évaluations sommatives

---

Exercices	
Description	Exercices pratiques réalisés en classe permettant l'intégration des éléments d'apprentissage.
Pondération	10%
Modalité particulière	Travail individuel
Date et durée	Cours 6. Durée variable en fonction des exercices
Critères d'évaluation	Respect des consignes spécifiques

Travail pratique #1	
Description	Développer un site en PHP avec la programmation orientée objet
Pondération	20 %
Modalité particulière	Travail individuel
Date et durée	Du cours 7 au cours 11
Critères d'évaluation	Respect des consignes spécifiques

Travail pratique #2	
Description	Développer un site Web dynamique simple basé sur un patron d'architecture Modèle-Vue-Contrôleur (MVC)
Pondération	20 %
Modalité particulière	Travail individuel
Date et durée	Du cours 12 au cours 17
Critères d'évaluation	Respect des consignes spécifiques

Travail pratique #3	
Description	Développer un site Web dynamique simple basé sur un patron d'architecture Modèle-Vue-Contrôleur (MVC), connecté à une base de données SQL et des interfaces Administrateur / Client.
Pondération	20 %
Modalité particulière	Travail individuel
Date et durée	Du cours 18 au cours 24
Critères d'évaluation	Respect des consignes spécifiques

Épreuve finale : Produire un site Web dynamique en utilisant un patron de conception orientée objet.	
Description	<b>Produire un site Web dynamique utilisant une base de données en exploitant un langage de programmation orienté objet et basé sur un patron de conception spécifique.</b>
Pondération	30 %
Modalité particulière <sup>1</sup>	<ul style="list-style-type: none"> <li>• À partir de médias fournis (images, gabarit de page Web, etc.) et d'un devis de production;</li> <li>• Individuellement;</li> <li>• <del>En laboratoire;</del></li> <li>• <del>Sous la supervision directe d'un professeur;</del></li> <li>• <del>Durée maximale de 3 heures à la fin de la session;</del></li> <li>• Documentation permise.</li> </ul>
Date et durée	Cours 25
Critères d'évaluation	Analyse adéquate du devis de production (10 %) Utilisation adéquate des éléments du langage pour traiter et afficher des données dynamiques (10 %) Application adéquate des principes de la programmation orientée objet (20 %) Application adéquate de patron de conception (15 %) Fonctionnement correct du produit réalisé (25 %) Application adéquate des tests (10 %) Qualité optimale de la programmation (10 %)

<sup>1</sup> En raison de la situation particulière de la présente session, des adaptations à l'épreuve finale pourront être proposés.

## 4. Calendrier des activités

### **1. Modéliser et programmer des classes à l'aide d'un langage de programmation web côté-serveur.**

EHPA (tous) et EHP2 (tous)

**Durée approximative** : 24 heures

#### **Descriptions de l'étape<sup>2</sup> :**

À la première étape, l'étudiant poursuit ses apprentissages de la programmation Web côté serveur et en y appliquant les principes de la programmation orientée objet déjà introduits dans les cours de programmation Web précédents. Il révise et approfondi les notions de base de la programmation orientée objet (classe, objet, encapsulation, propriété, méthode, héritage, etc) et apprend à les appliquer dans le contexte de la programmation côté-serveur. Il apprend à modéliser des classes répondants aux besoins du projet et à en faire le développement. Il apprend à réaliser des tests sur ses classes afin d'en assurer le fonctionnement et la qualité. Finalement, il approfondi ses apprentissages des classes disponibles dans l'environnement de programmation PHP. À la fin de cette étape, l'étudiant peut utiliser des classes natives ainsi que modéliser, réaliser et tester des classes simples permettant notamment l'accès aux données, la manipulation de requêtes et des réponses HTTP ou la connexion à la base de données.

#### **Éléments de la compétence :**

EHPA

1. Analyser la demande.
2. Programmer les éléments du site Web.
3. Résoudre les erreurs de programmation.
4. Vérifier la qualité de la programmation.
5. Documenter la programmation.
6. Gérer les versions de code source

EHP2

1. Traiter les requêtes d'un navigateur Web.
2. Interagir avec une base de données en-ligne.
3. Répondre aux requêtes d'un navigateur Web.

<b>Cours</b>	<b>Contenu détaillé</b>	<b>Remise et évaluations</b>
<b>1</b>	<b>Introduction à la programmation orientée objet (POO)</b> Présentation du plan de cours. Identification de la classe, des attributs, des méthodes et des relations Comprendre les différents types de relations : association, héritage. Les normes de documentation du langage de programmation orientée objet. <ul style="list-style-type: none"> <li>Comme analyser un diagramme de classes (cas pratique)</li> </ul>	
<b>2-5</b>	<b>Introduction à la POO en PHP:</b> Principes de la programmation orientée-objet <ul style="list-style-type: none"> <li>classe,</li> </ul>	

<sup>2</sup> Tiré du plan-cadre

- objet
- propriété
- méthode
- constructeur
- héritage
- polymorphisme

Mise en pratique - Retour à l'extension MySQLi et PDO

- Comprendre les classes et les méthodes

Classes et objets natifs du langage PHP

- Techniques de déclaration et de manipulation des classes et des objets du langage de programmation.
- Règles de syntaxe et de sémantique du langage PHP orienté objet
- Création d'une classe simple (propriétés et méthodes).
- Comment utiliser une classe (Instance) et récupérer les objets.
- Déclarer et utiliser les accesseurs (get) et les mutateurs (set).
- L'héritage et le polymorphisme.
- Classes dérivées, classes abstraites et interfaces.
- Les exceptions: gestion des erreurs en PHP.
- Méthodes de gestion des conflits

Atelier accompagné

## 6 Types et méthodes de tests

Gestion de test de logiciels

- Tests unitaires avec PHPUnit.
- Test d'intégration

Utilisation de outils de test PHP comme :

- PHPTESTER,
- PHP Sandbox.

**Exercice #1** : Codage de classes et tests de ces classes (10%)

## 7 Interagir avec une base de données MySQL

**Distribution TP1**

Classes natives et classes externes (bibliothèque) de gestion de

la base de données

- Connexion
- Opération CRUD

Différence entre les Classes PDO & MySQLi

Techniques d'optimisation et de validation des requêtes MySQL  
(requête préparé, sécurité, injection SQL, intégrité des données,  
etc)

**8 Cours atelier, production du TP1**



## 2. Développer un site Web dynamique simple suivant un patron de conception orientée objet.

EHPA (tous) et EHP2 (tous)

**Durée approximative** : 24 heures

### Descriptions de l'étape<sup>3</sup> :

À la deuxième étape, l'étudiant poursuivra ses apprentissages en programmation orientée objet côté-serveur en développant un site Web dynamique simple basé sur un patron de conception tel que Modèle-Vue-Contrôleur (MVC). D'abord, l'étudiant est introduit à divers patrons de conceptions et à leurs caractéristiques. Il voit comment appliquer de tels patrons et les contextes dans lesquels ceux-ci sont pertinents. Il sera aussi initié à divers types d'architectures de projet Web et aux diverses structures de réponses des serveurs tels que l'architecture trois tiers et les services Web. À cette étape, l'étudiant sera amené à développer un site Web dynamique basé sur une architecture commune en développement Web tel que MVC. Finalement, il poursuivra ses apprentissages des méthodes de validation de son code en mettant en place une suite de tests automatisés.

### Éléments de la compétence :

EHPA

1. Analyser la demande.
2. Programmer les éléments du site Web.
3. Résoudre les erreurs de programmation.
4. Vérifier la qualité de la programmation.
5. Documenter la programmation.
6. Gérer les versions de code source

EHP2

1. Traiter les requêtes d'un navigateur Web.
2. Interagir avec une base de données en-ligne.
3. Répondre aux requêtes d'un navigateur Web.

Cours	Contenu détaillé	Remise et évaluations
9	<b>Introduction au Modèle-Vue-Contrôleur (MVC)</b>  Comprendre le patron d'architecture MVC  Les concepts de base : Modèle, Vue, Contrôleur  Comparaison des techniques (forces et faiblesses)  Comment l'implémenter dans une approche orienté objet.	
10 - 15	<b>Patron d'architecture MVC</b>  Implémentation du patron d'architecture MVC dans un projet. Création de la Base de données / tables Modèles : <ul style="list-style-type: none"> <li>• Développer des classes pour se connecter et manipuler des données sur la base de données</li> <li>• Utiliser les méthodes pour réaliser des requêtes sur la base de données à l'aide des classes CRUD</li> </ul> Contrôleurs : <ul style="list-style-type: none"> <li>• Classes et Méthodes</li> <li>• Techniques avancées de réponse aux requêtes http (avec et sans données, développement des Vues,</li> </ul>	<b>Remise TP1 (cours 11)</b>  <b>Distribution TP2 (cours 12)</b>

<sup>3</sup> Tiré du plan-cadre

format des réponses tel que JSON, XML)

Vue :

- Le design et la récupération de données
- Installation une bibliothèque de code externe (TWIG), à l'aide de composer, pour faire la gestion du contenu de la vue.

Concept de Routeurs

Gestion du fichier .htaccess

Sécuriser le traitement des requêtes http

Atelier accompagné

**16 Cours atelier, production du TP2**

### **3. Développer un site Web dynamique avec une interface administrative suivant un patron de conception orientée objet.**

EHPA (tous) et EHP2 (tous)

**Durée approximative** : 27 heures

#### **Descriptions de l'étape<sup>4</sup> :**

À la troisième étape, l'étudiant appliquera les concepts vus auparavant et sera amené à développer un site Web dynamique incluant une interface administrative complète et personnalisée. À partir d'une architecture trois tiers et d'un patron de conception adéquat, l'étudiant développera les fonctionnalités de traitement des requêtes http, d'affichage des contenus, de mise à jour des données et de connexion des utilisateurs. Il s'assurera de la sécurité de son code et du fonctionnement adéquat de celui-ci en planifiant adéquatement les tests requis. Il sera introduit à l'installation, la configuration, la gestion et l'utilisation de bibliothèque externes de code en PHP. À la fin de cette étape, l'étudiant sera en mesure de développer et déployer un site Web dynamique complet suivant une architecture et un patron de conception spécifique. Il sera en mesure de créer une interface administrative complète permettant la gestion des données du site, de sécuriser son code et d'en assurer la qualité.

#### **Éléments de la compétence :**

EHPA

1. Analyser la demande.
2. Programmer les éléments du site Web.
3. Résoudre les erreurs de programmation.
4. Vérifier la qualité de la programmation.
5. Documenter la programmation.
6. Gérer les versions de code source

EHP2

1. Traiter les requêtes d'un navigateur Web.
2. Interagir avec une base de données en-ligne.
3. Répondre aux requêtes d'un navigateur Web.

<b>Cours</b>	<b>Contenu détaillé</b>	<b>Remise et évaluations</b>
<b>17 - 18</b>	<p><b>Interface Client - Administrateur (MVC)</b></p> <p>Comprendre l'arrière de l'interface client et de l'interface administrateur.</p> <p>Procédures et techniques pour générer une page de login:</p> <ul style="list-style-type: none"> <li>• Valider l'accès.</li> <li>• Gestion des données persistantes (session, cookie)</li> <li>• Sécurité des requêtes HTTP</li> <li>• Sécuriser le mot de passe</li> <li>• Sécuriser la session</li> </ul> <p>Gestion des privilèges des utilisateurs.</p> <p>Modélisez une base de données pour gérer l'accès au site.</p> <p>Atelier accompagné</p>	<p><b>Remise TP2 (Cours 17)</b></p> <p><b>Distribution TP3 (Cours 18)</b></p>

<sup>4</sup> Tiré du plan-cadre

<b>19 - 20</b>	<b>L'interface client</b>  Implémentation du patron d'architecture MVC <ul style="list-style-type: none"> <li>• Modèles : Classes et Méthodes</li> <li>• Contrôleurs : Classes et Méthodes</li> <li>• Vue : Design</li> </ul> Création d'une page pour saisir un nouvel utilisateur <ul style="list-style-type: none"> <li>• Validation de données</li> <li>• Sécuriser le mot de passe</li> </ul> Création page de login <ul style="list-style-type: none"> <li>• Validation de l'utilisateur</li> <li>• Création d'une session sécurisée</li> </ul> Atelier accompagné	
<b>21 - 22</b>	<b>L'interface administrateur</b>  Implémentation du patron d'architecture MVC <ul style="list-style-type: none"> <li>• Modèles : Classes et Méthodes</li> <li>• Contrôleurs : Classes et Méthodes</li> <li>• Vue : Design</li> </ul> Sécuriser et optimiser l'accès aux données  Gestion de privilèges.  Méthodes CRUD  Validation de données Atelier accompagné	
<b>23</b>	<b>Cours atelier, production du TP3</b>	
<b>24</b>	<b>Conclusion du cours</b> Révision et préparation pour l'épreuve finale.	<b>Remise TP3</b>
<b>25</b>	Épreuve finale	<b>Épreuve finale</b>

## 5. Matériel

---

Aucun livre obligatoire.  
Une clé USB.  
Papiers et crayons.

## 6. Médiagraphie

---

PHP.net, <https://www.php.net/>

## 7. Politiques d'évaluation des apprentissages

---

Conformément aux politiques institutionnelles et départementales d'évaluation des apprentissages, les modalités d'application spécifiques dans le contexte de la formation continue sont disponibles ici :

[http://fc.maisonneuve.qc.ca/sites/default/files/FC\\_Modalite\\_PIEA\\_582.pdf](http://fc.maisonneuve.qc.ca/sites/default/files/FC_Modalite_PIEA_582.pdf)

La Politique institutionnelle d'évaluation des apprentissages (PIEA) et la Politique départemental d'évaluation des apprentissages (PDEA) précisent les modalités d'application de l'ensemble des objectifs, principes et règles relatives à l'évaluation des apprentissages de tous les étudiants inscrits au Collège de Maisonneuve et spécifiquement des étudiants de Multimédia (discipline 582). La PIEA est accessible à partir de la page d'accueil d'Omnivox dans la section « Documents de référence » et la PDEA du département de multimédia est disponible avec le lien suivant : <https://goo.gl/QpB728>