

PLAN DE COURS

Titre et numéro : 420-SF1-RE Introduction à la programmation

Programme : Sciences, informatique et mathématique

Discipline: Informatique

Secteur : Enseignement régulier

Pondération : 2-3-3

Préalable : Aucun

Session : Automne 2023

Date : 21 août 2022 au 14 décembre 2023



Professeur	Bureau	Courriel	Contact
Maxime Barakatt	F-308	Voir contact	Teams en priorité, sinon Colnet
Patrice Roy	F-314	patrice.roy@clg.qc.ca	Colnet, Teams, 450-430-3120 poste 2780
Philippe Simard	F-318	philippe.simard@clg.qc.ca	Colnet, Teams
Coordonnateurs	Bureau	Courriel	Téléphone
Maxime Barakatt	F-308	maxime.barakatt@clg.qc.ca	Teams en priorité, sinon Colnet
Alexandre André-Lespérance	S-523	a.andre-lesperance@clg.qc.ca	Colnet, Teams, 450-420-3120 poste 2771

Présentation du cours

Ce cours a pour but d'initier les étudiantes et les étudiants au développement de programmes informatiques visant à résoudre des problèmes relativement simples.

Contribution du cours au programme

Ce cours est la base de la formation en informatique au SIM. Il est préalable à **tous** les autres cours de programmation de votre formation collégiale et, éventuellement, universitaire.

Les concepts présentés dans ce cours vous seront utiles à chaque fois que vous participerez au développement d'un programme informatique.

Compétence du devis ministériel

Le cours permet d'atteindre la compétence ministérielle (ONFA) « Développer des programmes informatiques pour résoudre des problèmes de nature scientifiques » dont la formulation se trouve en annexe à la fin de ce plan de cours.

Charge de travail

La pondération de ce cours est 2-3-3. Cela signifie qu'en plus des deux heures de cours théoriques et des trois heures de cours pratiques, l'étudiante ou l'étudiant doit s'attendre à fournir, **en moyenne**, trois heures de travail personnel par semaine à l'extérieur du cours.

Objectif d'intégration

À la fin de ce cours, l'étudiante ou l'étudiant sera capable de développer et coder, à l'aide d'un langage de programmation, des solutions algorithmiques à des problèmes de nature scientifique.

Objectifs d'apprentissage

- Planifier la résolution informatisée d'un problème de nature scientifique
- Élaborer un algorithme pour résoudre le problème.
- Exploiter un langage de programmation, ce qui signifie entre autres :
 - Utilisation d'un environnement intégré de développement d'un langage moderne : édition, compilation, exécution
 - Traduction des algorithmes en langage formel
 - Codage de fonctions; utilisation de divers types de passage de paramètres; utilisation de la valeur de retour d'une fonction
 - Utilisation des variables de type valeur et de type référence
 - Instanciation d'un objet et utilisation de ses méthodes
 - Instanciation et utilisation des tableaux
 - Introduction à la conception, la déclaration et l'utilisation d'une classe simple. la fin de ce cours, l'étudiante ou l'étudiant sera capable de développer et coder, à l'aide d'un langage de programmation, des solutions algorithmiques permettant de résoudre des problèmes relativement simples
- Vérifier le bon fonctionnement du programme

Évaluation

L'évaluation des apprentissages se fera à l'aide :

- d'environ une dizaine d'exercices et de laboratoires qui seront des travaux à faire sur une base quasi hebdomadaire;
- d'un test théorique qui aura lieu vers le milieu de la session
- d'une production finale d'intégration composée d'une partie théorique (un examen synthèse) et d'une partie pratique (un travail de fin de session).

Le ou vers le	Type d'évaluation	Pondération
Semaine 8 16 au 20 octobre	Test théorique de mi-session	20
Semaines 13 à 15 27 nov. au 14 déc.	PFI volet pratique	15
Semaine 15 11 au 14 décembre	PFI volet théorique (examen final)	20
<i>Durant la session</i>	Exercices et laboratoires – il y aura environ une dizaine de travaux à remettre	45
Total :		100

Production finale d'intégration

La production finale d'intégration comporte un volet théorique et un volet pratique. La pondération du volet pratique est égale à la pondération du volet théorique et le poids total des deux évaluations est conforme aux critères de la PIEA.

Plus précisément, le volet pratique de la production finale d'intégration doit vérifier la capacité **individuelle** de l'étudiant de compléter le code d'une application informatique relativement complexe en y ajoutant les parties qui apportent une solution algorithmique aux sous-problèmes plus simples à résoudre.

Par ailleurs, le volet théorique de la production finale d'intégration doit vérifier la compréhension théorique des concepts enseignés durant toute la session et la capacité individuelle de l'étudiant à écrire des algorithmes pour résoudre des problèmes simples et les coder sans l'aide de l'ordinateur. Ce volet de la production finale doit se faire sous la surveillance d'un enseignant ou d'une personne désignée et pourra, par exemple, prendre la forme d'un examen écrit.

Critères d'évaluation de la production finale d'intégration

Pour la production théorique

- qualité de la compréhension des concepts théoriques;
- précision des réponses;
- qualité de la présentation écrite.

Pour la production pratique

- qualité de l'exécution du programme;
- précision de l'application produite relativement au problème donné;
- qualité de la codification et de la conception.

Règles régissant l'évaluation

1. Sauf circonstances exceptionnelles dont le professeur est seul juge, l'absence à un examen entraîne la note zéro pour cet examen. Il n'y a pas d'examen de reprise. Si le professeur juge qu'il y a des circonstances exceptionnelles, il pourra proposer à l'étudiant un arrangement individuel ou encore imposera des conditions de réussite particulières. Chaque cas sera analysé au mérite.

C'est à l'étudiante ou à l'étudiant qu'il revient de rencontrer son enseignante ou son enseignant pour lui faire part, dès son retour au Collège, des motifs de son absence et lui fournir une pièce justificative. Dans le cas d'un motif exceptionnel et justifié, l'enseignante ou l'enseignant pourra proposer à l'étudiante ou à l'étudiant une modalité de reprise ou une modification de barème.

2. Sauf circonstances exceptionnelles dont le professeur est seul juge, tout travail en retard **ne sera pas corrigé** et aura droit à la note **zéro**.

La politique institutionnelle d'évaluation des apprentissages (PIEA) prévoit des dispositions en cas de plagiat. Voici un extrait de certains articles qui touchent le plagiat et la tricherie.

6.7 Le plagiat et la tricherie

Par la présente politique, le Collège déclare que le plagiat et la tricherie, sous toutes leurs formes, sont contraires aux valeurs qui orientent sa mission. Elles constituent des infractions graves qui, lorsque constatées, entraînent des pénalités.

6.7.1 Les définitions

« [L']utilisation frauduleuse de l'œuvre d'autrui soit par emprunt, soit par imitation » constitue un plagiat. **L'étudiante ou l'étudiant qui permet que son travail soit copié en tout ou en partie est réputé aussi coupable que le plagiaire.**

Le plagiat comprend :

- a) la présentation ou la remise du travail d'une autre personne comme étant le sien;
- b) le copiage en tout ou en partie ou la traduction de l'œuvre d'autrui (publiée ou non publiée) sans en indiquer la source;
- c) la paraphrase de l'œuvre d'autrui (publiée ou non publiée) sans en indiquer la source, à moins que cela ne constitue l'objet d'une activité d'évaluation formative ou sommative.

Toute autre forme de pratique malhonnête ou de dissimulation lors d'une évaluation sommative est considérée comme une tricherie. La tricherie comprend, notamment :

- a) la falsification d'un rapport de laboratoire;
- b) la préparation d'un travail pour quelqu'un d'autre ou la réalisation de son travail par quelqu'un d'autre;
- c) le fait de prétendre avoir remis un travail qui en fait n'a jamais été remis;
- d) la soumission de fausses informations, par exemple un faux certificat médical, afin d'obtenir un délai dans la remise d'un travail, dans le but d'éviter des sanctions ou d'obtenir un privilège;
- e) le fait de demander à quelqu'un d'autre de signifier sa présence en classe alors que l'on est absent;
- f) le fait d'attester la présence en classe d'une autre étudiante ou d'un autre étudiant absent;
- g) l'obtention ou la tentative d'obtention, au cours de la réalisation d'une activité d'évaluation, d'une aide de quelque nature que ce soit sans y être autorisé;
- h) l'aide ou l'assistance à une autre étudiante ou à un autre étudiant, au cours de la réalisation d'une activité d'évaluation, sans y être autorisé;
- i) l'emploi ou la possession de tout matériel, de tout instrument ou de tout moyen pouvant servir au stockage, à la diffusion ou à la recherche d'informations sans y être autorisé;
- j) le fait de réaliser toute forme d'évaluation à la place de quelqu'un d'autre;
- k) le fait de confier la réalisation de ses propres évaluations à quelqu'un d'autre;
- l) la communication, lors d'une activité d'évaluation en classe, avec quelqu'un d'autre, de quelque façon que ce soit, sans y être autorisé.

Le plagiat et la tricherie peuvent être décelés par l'enseignante ou par l'enseignant avant, pendant ou après la tenue de l'évaluation.

Tout plagiat, tentative de plagiat ou collaboration à un plagiat ou à une tricherie entraîne la note de zéro pour le travail ou l'examen en cause, et entraîne les conséquences prévues aux articles 6.7.2 et 6.7.3 de la PIEA¹.

3. La réussite du cours est conditionnelle à la réussite de la production finale d'intégration. Le seuil de réussite est fixé globalement à 60 % pour cette activité.
4. Un travail peut voir sa note varier jusqu'à 10 % en plus ou en moins en raison de l'excellence ou la déficience du français lorsqu'applicable.
5. Pour toute autre situation non précisée dans ce plan de cours, la PIEA s'applique.

Matériel obligatoire à se procurer

Vous devez vous procurer le matériel suivant pour ce cours :

1. Un dispositif vous permettant de conserver copie de vos travaux. Il peut s'agir d'une clé USB, d'un disque dur physique ou d'un espace disque dans le nuage (Google Drive, One Drive, DropBox, etc.). N'avoir qu'une seule copie d'un travail vous expose à des problèmes éventuels – tôt ou tard, vous risquez de perdre un travail par inadvertance ou malchance, généralement la veille ou le matin de la remise.
2. Les travaux de plus d'une page doivent obligatoirement être remis agrafés si une version papier a été demandée. Une **agrafeuse** est donc requise².
3. Des feuilles mobiles ainsi que des stylos, pousse-mines ou crayons à la mine permettant de prendre des notes durant les cours³. Une gomme à effacer est requise puisque vous ne pourrez pas demander à votre voisin de vous prêter sa gomme à effacer durant les tests et les examens.

¹ Le texte complet de la PIEA se trouve au <http://www.clg.qc.ca/fileadmin/clg/publication/reglement/PIEA.pdf>

² Et des points seront enlevés à votre travail s'il n'est pas agrafé. Il y a normalement une agrafeuse sur la table du professeur dans le laboratoire mais il est déjà arrivé qu'elle *disparaisse mystérieusement*.

³ Vous pourrez procéder comme bon vous semble pour les notes de cours mais une approche traditionnelle est toujours une bonne idée. On a encore des papyrus de l'Égypte de la grande époque; vous avez souvent de la difficulté à retrouver dans vos affaires un fichier créé le mois dernier!

Contenu

Durant la session, nous aborderons le contenu suivant :

- 1 Planifier la résolution informatisée d'un problème de nature scientifique
 - Identifier les concepts impliqués dans un problème
 - Identifier les intrants, extrants et traitements
 - Identifier les types de données pertinents
 - Préparation d'un jeu d'essais en vue de valider le fonctionnement du programme

- 2 Élaborer un algorithme pour résoudre le problème
 - Découpage du problème en sous-problèmes
 - Représenter le découpage fonctionnel de la solution
 - Représenter l'algorithme sous une forme neutre
 - Utilisation de structures de contrôle
 - Utilisation d'expressions arithmétiques, relationnelles et logiques
 - Réaliser la trace d'un algorithme
 - Concevoir l'algorithme avec un souci d'efficacité

- 3 Exploiter un langage de programmation
 - Traduire un algorithme en langage formel dans le respect de la syntaxe, des standards et des bonnes pratiques de programmation
 - Comprendre les concepts entourant les constantes littérales et symboliques
 - Comprendre la déclaration, la portée et l'initialisation des variables
 - Utiliser les différents types de données
 - Instancier et utiliser des objets
 - Utiliser des fonctions
 - Utiliser des bibliothèques standards
 - Documenter de façon pertinente le code
 - Comprendre les concepts abstraits de pile et de tas
 - Utiliser et déclarer des énumérations

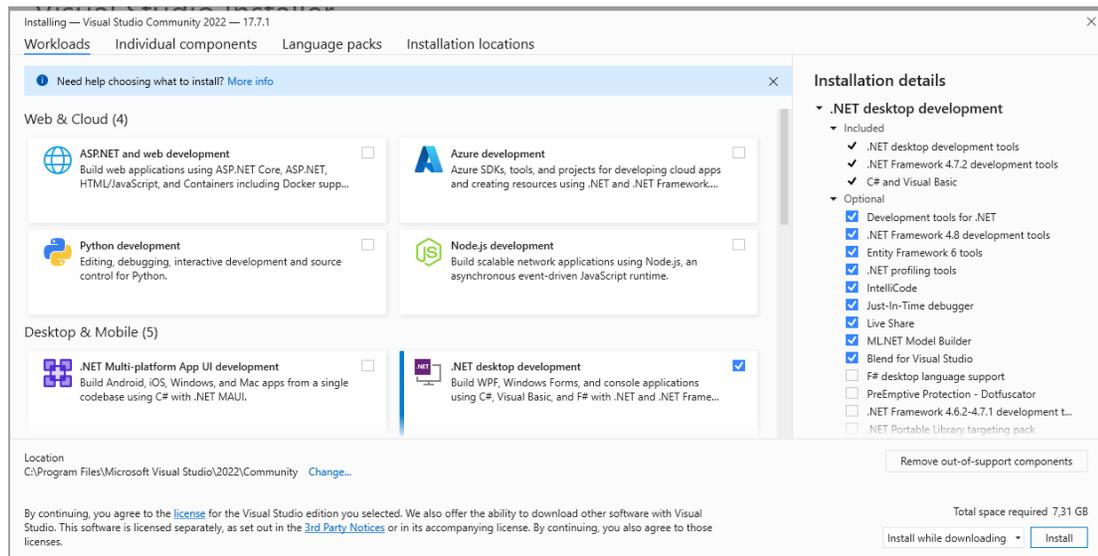
- 4 Vérifier le bon fonctionnement du programme
 - Comprendre les différents messages d'erreur de l'environnement intégré de développement et du compilateur
 - Utiliser l'outil de développement pour déboguer

Sur le plan global, nous prendrons soin d'utiliser le système de fichiers pour organiser nos projets et d'utiliser l'environnement intégré de développement d'un langage moderne.

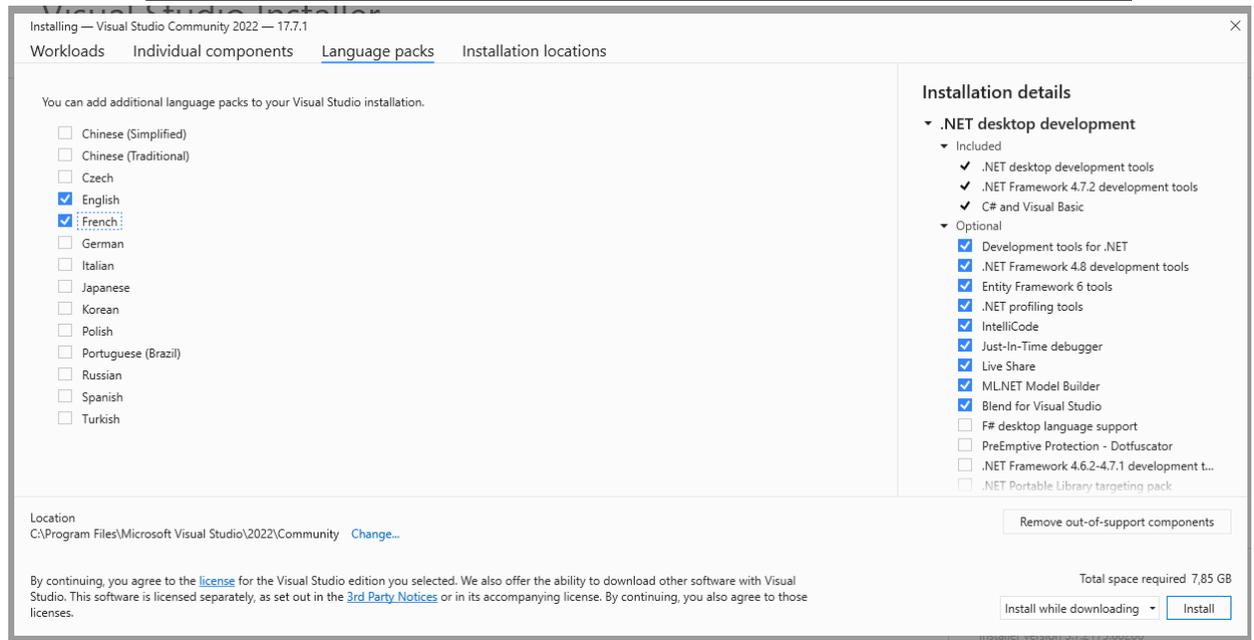
Méthodologie

Le système de développement *Visual Studio Community 2022* servira d'outil pour les applications pratiques. Nous utiliserons le langage C# de Microsoft comme outil de développement. Vous pouvez télécharger gratuitement ce logiciel en cliquant sur le lien suivant puis sur le bouton *télécharger*. Suivez ces étapes :

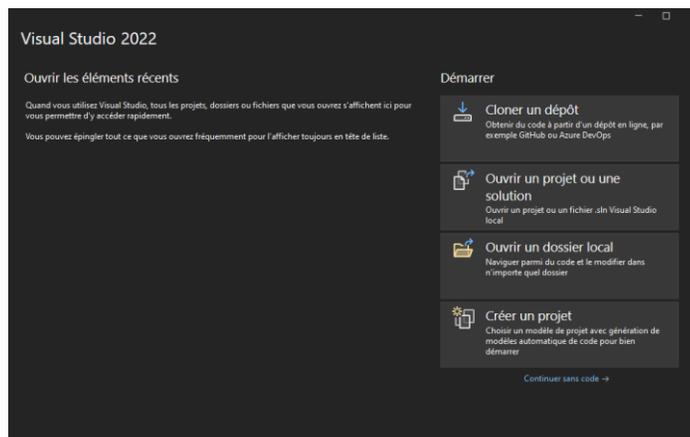
1. Exécutez le fichier téléchargé *VisualStudioSetup.exe*
2. Une fois l'installateur lancé, la fenêtre suivante s'affichera. Sélectionnez *.NET desktop développement* (ou l'équivalent français selon la langue de votre installateur).



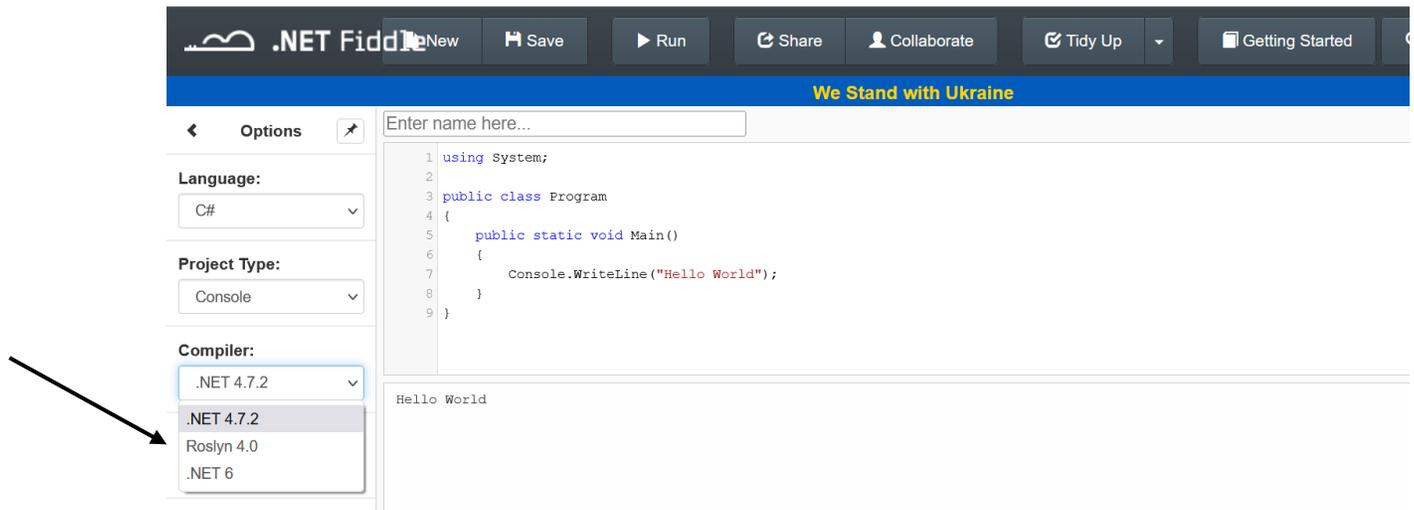
3. Dans l'onglet *Language packs*, sélectionnez *French*



4. Cliquez ensuite sur *Install* en bas à droite.
5. Une fois l'installation terminée, on vous demandera de vous connecter. Comme adresse courriel vous devez entrer NUMÉRODADMISSION@edu.clg.qc.ca et comme mot de passe, utilisez le même que pour le Wi-Fi du collège.
6. Il est possible que Visual Studio soit lancé automatiquement, autrement vous pouvez le lancer manuellement à l'aide du menu Démarrer. Vous devriez alors avoir la fenêtre suivante.



Si vous le souhaitez, il est également possible d'utiliser l'émulateur qui se trouve au <https://dotnetfiddle.net/T3cU96>, par exemple si vous utilisez une machine sur laquelle *Visual Studio Community 2022* n'est pas installée. Ce raccourci vous mènera vers un programme « vide » .NET 6. Si vous passez directement par le lien plus général qu'est <https://dotnetfiddle.net/s> alors prenez soin de choisir le compilateur **.NET 6** dans la section gauche de l'affichage.



Procédure de remise des travaux

La version électronique des travaux et exercices devra être remise selon la procédure qui sera expliquée en classe par le professeur; il pourra s'agir d'une boîte de remise de Colnet ou Teams ou tout autre dispositif préféré par votre enseignant.

Médiagraphie

Les documents requis durant le cours, que ce soient les notes de cours théoriques, les laboratoires ou les exemples, seront disponibles via Colnet à l'onglet **Ressources** de la rubrique **Cours**. Votre enseignant pourra toutefois décider de procéder autrement et passer par son propre site Web ou d'un dossier partagé sur le *nuage*.

Le site <https://h-deb.clg.qc.ca/> est également un site de référence intéressant pour le cours. Il s'agit d'un site généraliste sur la programmation dont la pertinence vous semblera de plus en plus évidente au fur et à mesure que vous progresserez en informatique. Un gros merci à Patrice Roy de l'alimenter et le mettre à jour régulièrement.

Démarche officielle du Cégep pour les conflits enseignants / étudiants

En cas de conflit avec un enseignant, il est important de suivre la démarche suivante :

1. Dans un premier temps, il faut en parler **directement avec l'enseignant concerné** et miser sur le dialogue;
2. Dans le cas où le différend ne se règle pas, l'étudiant s'adresse **au coordonnateur du département** concerné et tente de régler le problème avec son aide;
3. **Si le problème persiste toujours**, l'étudiant peut rencontrer son API au local L-117, qui analysera la situation problématique avec lui et qui pourra lui suggérer de déposer une plainte en remplissant le formulaire à cet effet. La plainte sera acheminée au directeur adjoint de la Direction des études qui verra à trouver une solution.

Voir : <http://www.clg.qc.ca/cheminement-registrariat/conflit-entre-un-enseignant-et-un-etudiant/>

Annexe A – Compétence 02NFA

Code : 0NFA	
<i>Objectif</i>	<i>Standard</i>
Énoncé de la compétence Développer des programmes informatiques pour résoudre des problèmes de nature scientifique.	Critères de performance liés à l'ensemble de la compétence <ul style="list-style-type: none"> • Utilisation appropriée de la terminologie. • Utilisation correcte des outils de développement requis. • Démonstration de rigueur et de persévérance.
Éléments de la compétence	Critères de performance
1. Planifier la résolution informatisée d'un problème de nature scientifique.	<ul style="list-style-type: none"> • Reconnaissance appropriée des concepts impliqués dans un problème. [1] • Décomposition correcte du problème. • Détermination juste des intrants et des extrants. • Reconnaissance juste des types de données de base et composées. [2] • Détermination juste des traitements nécessaires. • Préparation d'un jeu d'essais approprié en vue de valider le fonctionnement du programme.
2. Élaborer un algorithme pour résoudre le problème.	<ul style="list-style-type: none"> • Structuration cohérente de l'algorithme jusqu'aux instructions. • Organisation efficace des instructions. [3] • Utilisation appropriée des types de données et des tableaux. • Utilisation juste des expressions arithmétiques, logiques et relationnelles. • Réalisation correcte de la trace d'exécution d'un algorithme. • Conception d'un algorithme correct dans un souci d'efficacité.
3. Exploiter un langage de programmation.	<ul style="list-style-type: none"> • Traduction conforme de l'algorithme dans le langage de programmation requis. • Respect de la syntaxe du langage de programmation. • Application rigoureuse des standards et des bonnes pratiques de programmation. • Utilisation appropriée de bibliothèques standards. • Documentation pertinente du code informatique. • Utilisation appropriée des types de données et de leurs comportements, s'il y a lieu.
4. Vérifier le bon fonctionnement du programme.	<ul style="list-style-type: none"> • Repérage des erreurs de fonctionnement. • Reconnaissance juste de messages. [4] • Utilisation correcte d'un débogueur. • Pertinence des correctifs.