



PLAN DES ÉTUDES GYMNASIALES

DOMAINE INFORMATIQUE INFORMATIQUE, DISCIPLINE OBLIGATOIRE

1. Dotation horaire (leçons hebdomadaires)

Niveaux	1	2	3	4
Discipline fondamentale	2	2	-	-

2. Objectifs généraux de formation

L'enseignement de l'informatique amène les élèves à développer des compétences transversales, tout en familiarisant les élèves avec les concepts généraux qui sous-tendent le traitement automatique des données et qui peuvent être transférés à des applications actuelles ou futures.

Sur les plans de la théorie et de la pratique, l'enseignement de l'informatique permet d'acquérir des compétences universelles telles que les stratégies de résolution de problèmes, le raisonnement structuré et le travail de précision ; il laisse cependant également la place à la créativité et ouvre de nouvelles perspectives de conception. Il fournit en outre une compréhension approfondie de la manière dont interagissent le monde réel et le monde virtuel, discute les avantages et inconvénients des systèmes informatisés pour la science ainsi que pour la société et apprend à utiliser avec efficacité, tout en faisant preuve d'esprit critique, les solutions logicielles. L'enseignement de l'informatique apporte ainsi une importante contribution à l'aptitude générale aux études supérieures ainsi qu'au développement personnel des élèves.

3. Objectifs fondamentaux

La discipline fondamentale informatique a pour objectif de développer chez l'élève les savoirs, savoir-faire et savoir-être (attitudes) ci-dessous.

3.1. Connaissances

- Comprendre certains principes fondamentaux de l'informatique (traitement, stockage, communication de l'information, etc.)
- Connaître différentes représentations et structurations de l'information
- Connaître les fondements d'un langage de programmation
- Connaître les risques inhérents à l'utilisation des technologies numériques

3.2. Savoir-faire

- Analyser des problèmes et développer des solutions correspondantes
- Retranscrire les solutions à l'aide de moyens informatiques (algorithmes, programmes, bases de données, etc.)
- Adopter de bonnes pratiques en matière de sécurité et de protection des données et de la sphère privée
- Reconnaître les liens entre théorie et pratique

3.3. Savoir-être (attitudes)

- Faire preuve d'un esprit critique constructif face aux réalisations techniques
- Faire preuve de persévérance et de rigueur

- Faire preuve de curiosité, de créativité et d'autonomie dans l'expérimentation et la conception
- Être sensible aux implications de l'informatique dans la vie quotidienne
- Être conscient de la diversité de solutions possibles
- Faire preuve d'esprit de collaboration

4. Objectifs sommaires, contenus, matières apparentées

Prérequis. Ce cours n'a pas de prérequis spécifique.

Première année

Objectifs sommaires	Contenus	Matières apparentées
	1. Algorithmes et programmation	
<ul style="list-style-type: none"> - Définir ce qu'est un algorithme - Représenter un algorithme de différentes manières - Interpréter des algorithmes simples - Savoir utiliser les principales méthodes de résolution de problèmes - Résoudre des problèmes simples à l'aide d'algorithmes 	1.1 Algorithmes <ul style="list-style-type: none"> - Pseudocode, ordinogramme - Exécution pas à pas, tableau d'états - Méthodes de résolution de problèmes : spécification du problème, décomposition en sous-problèmes, amélioration par itération, passage du simple au complexe, abstraction, vérification et validation 	Langues Philosophie Économie Mathématiques
<ul style="list-style-type: none"> - Se servir des éléments essentiels d'un langage de programmation 	1.2 Programmation <ul style="list-style-type: none"> - Instruction, expression, séquence, structures de contrôle, imbrication, constante, variable, procédure, commentaire - Nombre, booléen, texte 	Mathématiques Physique Biologie Chimie Économie Langues Arts visuels Musique
	2. Représentation de l'information	
<ul style="list-style-type: none"> - Connaître les unités de mesure de l'information et les comparer - Faire la distinction entre informations analogiques et numériques - Comprendre la représentation binaire des nombres entiers et effectuer des opérations simples (conversion, etc.) - Comprendre la représentation binaire des caractères et des images 	<ul style="list-style-type: none"> - Bit, octet et leurs multiples - Limitation des types de données - Encodage de caractères, de nombres et d'images - Image vectorielle, image bitmap - Informations analogiques et informations numériques 	Langues Mathématiques Physique Chimie Biologie Géographie Arts visuels Musique

	3. Architecture	
<ul style="list-style-type: none"> - Décrire l'architecture d'un ordinateur et le fonctionnement de ses composants principaux - Expliquer l'interaction entre le matériel (hardware), le système d'exploitation et les applications 	3.1 Ordinateur <ul style="list-style-type: none"> - Modèle en couches (matériel, firmware, système d'exploitation, pilotes, applications, etc.) 	Physique Biologie
	4. Web	
<ul style="list-style-type: none"> - Comprendre les concepts de base des langages HTML et CSS - Utiliser une documentation de référence pour créer une page web simple - Analyser la structure de pages web existantes 	<ul style="list-style-type: none"> - Site web - Structure d'une page HTML - Différence entre forme et contenu - Éléments HTML (balises et attributs) - Classes, identifiants et propriétés CSS - URL 	Langues Économie Droit Arts visuels
	5. Stockage de données	
<ul style="list-style-type: none"> - Comprendre le concept de fichier - Comprendre les concepts de dossier et d'arborescence de fichiers 	5.1 Fichiers et formats <ul style="list-style-type: none"> - Structure d'un fichier : nom, extension, métadonnées - Types de fichiers (texte, binaire, etc.) - Encodage des caractères 	Mathématiques Économie
	6. Aspects liés à la société de l'information	
<ul style="list-style-type: none"> - Prendre des mesures de protection appropriées 	6.1 Protection des données <ul style="list-style-type: none"> - Identité numérique officielle et non officielle (traces, cookies, agrégation de profils, etc.) - Authentification à multiples facteurs - Qualité de mots de passe sur le plan de la sécurité 	Mathématiques Droit Philosophie Psychologie
<ul style="list-style-type: none"> - Comprendre les enjeux personnels et citoyens de la récolte et de l'exploitation de données massives - Respecter les aspects principaux des droits et obligations dans l'utilisation des données 	6.2 Aspects citoyens <ul style="list-style-type: none"> - Données récoltées lors de l'utilisation d'Internet et agrégation des données - Impacts sociétaux de l'utilisation des technologies de l'information (notamment du cloud et des réseaux sociaux) - Droit d'auteur (licences Creative Commons), droit à l'image 	Économie Droit Psychologie Histoire Musique Arts visuels Langues

<ul style="list-style-type: none"> - Reconnaître les risques de sécurité et les expliquer - Comprendre l'utilité et les principes des connexions sécurisées 	<p>6.3 Sécurité</p> <ul style="list-style-type: none"> - Virus et autres menaces - Social engineering 	<p>Mathématiques Économie Biologie Droit Psychologie Histoire</p>
---	--	---

Deuxième année

Objectifs sommaires	Contenus	Matières apparentées
	<p>1. Algorithmes et programmation</p>	
<ul style="list-style-type: none"> - Comparer des algorithmes simples - Prendre conscience de la différence d'efficacité des algorithmes pour résoudre un même problème - Prendre conscience des limites de l'algorithmique pour certains problèmes 	<p>1.1 Algorithmes</p> <ul style="list-style-type: none"> - Approfondissement des contenus étudiés en 1^{ère} année 	<p>Langues Philosophie Économie Mathématiques</p>
<ul style="list-style-type: none"> - Diagnostiquer et corriger une erreur - Comprendre les types de données de base - Utiliser les types de données appropriés pour des problèmes simples 	<p>1.2 Programmation</p> <ul style="list-style-type: none"> - Fonction - Erreur de syntaxe, d'exécution et de sémantique - Tableau, liste 	<p>Mathématiques Physique Biologie Langues Arts visuels Musique</p>
	<p>2. Représentation de l'information</p>	
<ul style="list-style-type: none"> - Comprendre les principes permettant de détecter et de corriger des erreurs 	<ul style="list-style-type: none"> - Code de correction d'erreur - Compression de données 	<p>Langues Mathématiques</p>
	<p>3. Architecture</p>	
<ul style="list-style-type: none"> - Expliquer la différence entre réseau local et réseau global - Expliquer le paradigme client-serveur - Expliquer les implications d'une communication sécurisée et d'une communication non sécurisée 	<p>3.2 Réseaux</p> <ul style="list-style-type: none"> - LAN, WAN, web, nom de domaine (DNS) et adresse IP, routage - Protocoles http et https, réseau sans fils - Cloud (stockage) : aspects techniques 	<p>Physique Biologie Mathématiques</p>
	<p>4. Web</p>	
<ul style="list-style-type: none"> - Comprendre le concept d'application web - Comprendre l'interaction d'une page web avec un serveur web et une base de données 	<ul style="list-style-type: none"> - Application web - Distinction exécution côté serveur et côté client 	<p>Langues Économie Droit Arts visuels</p>

	5. Stockage de données	
<ul style="list-style-type: none"> - Comprendre la notion de format structuré et semi-structuré - Comprendre la notion de stratégie de sauvegarde 	5.1 Fichiers et formats <ul style="list-style-type: none"> - Formats structurés (CSV, etc.) et semi-structurés (XML, JSON, etc.) - Sécurité des données (stratégie de sauvegarde) 	Mathématiques Économie
<ul style="list-style-type: none"> - Modéliser et comprendre les enjeux à l'aide d'exemples - Analyser et écrire des requêtes 	5.2 Base de données <ul style="list-style-type: none"> - Base de données relationnelles 	Mathématiques Économie Droit
	6. Aspects liés à la société de l'information	
<ul style="list-style-type: none"> - Comprendre les applications du chiffrement dans l'informatique d'aujourd'hui - Comprendre les principes de fonctionnement du chiffrement et du déchiffrement - Chiffrer des données à l'aide d'un outil existant actuel 	6.3 Sécurité <ul style="list-style-type: none"> - Concepts clefs du chiffrement : symétrique, asymétrique, chiffrement, déchiffrement, signature (authentification), non-répudiation, non-falsification, fonction de hachage, etc. - Chiffre monoalphabétique (par exemple chiffre de César) 	Mathématiques Économie Biologie Droit Psychologie Histoire

5. Indications méthodologiques et didactiques

- L'informatique scientifique est une branche dans laquelle l'apprentissage se fonde essentiellement sur la pratique (*learning by doing*). Elle nécessite l'utilisation par l'élève d'un ordinateur chaque semaine. Il est donc nécessaire que chaque élève dispose d'un ordinateur pour toutes les périodes du cours.
- Les thématiques seront abordées à l'aide de sujets qui touchent directement le quotidien des élèves et leurs préoccupations.
- Dans la mesure du possible, les notions de programmation devraient dans un premier temps être introduites de manière indépendante d'un langage de programmation, par exemple au moyen d'un langage de programmation visuel.
- Au-delà de sa présence en tant qu'objet d'apprentissage, la robotique peut jouer un rôle déterminant en tant qu'outil transversal d'apprentissage dans de nombreux thèmes du plan d'études informatique, notamment la programmation, l'algorithmique, etc.
- De manière générale, les thèmes peuvent être traités parallèlement, en illustrant les concepts par des applications pratiques issues d'autres thèmes.
- Lorsque le thème s'y prête, les notions seront abordées avec une approche historique.
- Le travail en projets sera favorisé, en donnant l'occasion d'aborder une méthodologie de projet.

6. Enseignement interdisciplinaire

L'informatique est présente dans presque tous les aspects de la vie. Les possibilités d'enseignement interdisciplinaires existent avec toutes les branches, par exemple :

- arts visuels : publication sur Internet, graphiques, présentations
- langues : communication, documentation, linguistique et sémantique, traduction automatique
- philosophie, histoire, psychologie : éthique, aspects sociaux et sociétaux de l'informatique, cognition
- économie & droit : droit de l'informatique, informatique de gestion, modèles économiques, simulations démographiques
- mathématiques, physique : procédures numériques, électronique, simulations de processus aléatoires, simulations de chocs élastiques
- sciences naturelles : mesures et traitements de données, logiciels de modélisation, simulations de systèmes dynamiques
- projets interdisciplinaires impliquant de la gestion de projet et des processus de modélisation

Plan d'études discipline informatique S2

Ce document accompagne le plan d'études « Informatique » élaboré par le groupe de travail « Informatique au S2 », mandaté par la CORECOFR et le S2 de la DICS.

Objectifs généraux de formation et objectifs fondamentaux

La discipline informatique, parallèlement aux aspects scientifiques, inclut également dans ses objectifs d'apprentissage les aspects sociétaux et juridiques découlant de la numérisation de la société, conformément à l'objectif stratégique 3 de la stratégie numérique du 21 juin 2018 de la CDIP : « s'assurer que les élèves et les personnes en formation acquièrent les compétences nécessaires pour gérer la numérisation et les aider à devenir des citoyennes et citoyens autonomes et responsables dans un monde numérique ».

Objectifs sommaires, contenus, matières apparentées

Les objectifs sommaires, contenus et matières apparentées sont déclinés en deux parties, correspondant aux objectifs et contenus à aborder respectivement durant la première et la deuxième année de gymnase.

Les objectifs sont en outre groupés en 6 thèmes, chacun des thèmes étant abordé au cours des deux années, dans une logique d'approfondissement.

En raison de la spécificité de la discipline informatique, le PE est conçu de façon à ce que les objectifs et compétences de thèmes différents de la même année puissent être étudiés parallèlement, lorsque cela est judicieux. Par exemple, il peut être intéressant d'illustrer la notion d'algorithme, étudiée dans le thème 1, à l'aide d'éléments étudiés dans le thème 2, représentation de l'information.

Le tableau ci-dessous indique le nombre indicatif de périodes dédiées, par thème et par année. Ce découpage tient compte d'un nombre réaliste de 60 périodes d'enseignement annuelles, une fois soustraites les périodes d'évaluations, tombant sur des jours de congés ou des activités spéciales de l'école.

Nombre de périodes par thème et par année						
Thème	1ère année		2e année		Total	
	Min	Max	Min	Max	Min	Max
Algorithmes et programmation	24	30	22	24	46	54
Représentation de l'information	4	6	4	6	8	12
Architecture	4	6	6	8	10	14
Web	8	10	8	12	16	22
Stockage de données	2	4	8	12	10	16
Aspects liés à la société de l'information	10	12	6	8	16	20
Totaux	52	68	54	70	106	138
Maximum	60		60		120	

Compétences bureautiques

Les compétences en bureautique¹ ne font pas l'objet d'objectifs d'apprentissage de la discipline informatique et ne seront donc pas évaluées dans le cadre de ce cours.

Le plan d'études aborde l'étude de ces compétences au travers des thèmes décrits ci-dessus. Les outils bureautiques seront utilisés à chaque fois qu'il sera possible, dans la mesure où ils apportent une plus-value didactique et pédagogique. En particulier, l'utilisation du tableur pourra contribuer avec profit à certains thèmes.

Le constat du niveau insuffisant de la plupart des élèves dans les compétences en bureautique montre qu'il est nécessaire d'organiser l'apprentissage de ces compétences au degré secondaire 2, dans un cadre distinct de la discipline informatique. Les modalités pertinentes pourraient être de différents types, notamment : modules de soutien offerts durant la rédaction du travail de maturité, cours en ligne à suivre en autonomie, cours facultatifs, tutorat par les pairs, etc. Une réflexion est nécessaire pour approfondir cette question.

De plus, il est à noter que ces compétences peuvent et doivent être abordées de manière transversale dans les autres disciplines du plan d'études gymnasiales, en complément de ce dispositif.

Matériel

Les apprentissages en informatique se fondent essentiellement sur la pratique et nécessitent l'utilisation par l'élève d'un ordinateur de manière presque permanente. Il est donc nécessaire que chaque élève dispose d'un ordinateur pour toutes les périodes du cours, même si des activités débranchées peuvent aussi être mises en œuvre.

Il s'agit ici d'un ordinateur complet, par opposition à une tablette. Ce dernier type d'appareil ne permet en effet pas de travailler de manière adéquate, notamment pour la branche informatique. Par ailleurs, une fois leur maturité en poche, au cours de leurs études universitaires, les étudiants utiliseront des ordinateurs, et il convient donc de les préparer à travailler efficacement avec ces appareils.

Travail en demi-classes

Afin de favoriser l'étude approfondie de certains thèmes ou le travail par projet, une heure par mois est dédoublée en demi-classes, de sorte à créer des ateliers.

Deux modèles possibles sont proposés ci-dessous, parmi d'autres. L'école détermine le modèle qui lui convient suivant ses spécificités en termes de locaux.

1. Modèle 1

Durant environ 15 semaines (sans compter les périodes d'évaluation) :

- Chaque semaine, les deux demi-classes ont ensemble 1 période, plus chacune 1 seconde période, à des horaires différents.

Durant environ 15 semaines (sans compter les périodes d'évaluation) :

- Chaque semaine, les deux demi-classes ont ensemble 2 périodes groupées.

¹ Compétences d'utilisation en traitement de texte, tableur, présentation, multimédia.

2. Modèle 2

Durant toute l'année scolaire, le cours se déroule selon un rythme de 4 semaines² :

- Semaine 1 et 2 : les demi-classes A et B ont ensemble 2 périodes ;
- Semaine 3 : les demi-classes A et B ont ensemble 1 période, la demi-classe A a de plus 2 périodes groupées ;
- Semaine 4 : les demi-classes A et B ont ensemble 1 période, la demi-classe B a de plus 2 périodes groupées.

Langage de programmation

Selon la décision de la CORECOFR, des langages de programmation communs à tous les établissements seront utilisés.

Pour l'apprentissage des concepts fondamentaux de la programmation, on utilisera un langage graphique basé sur la technologie Blockly tel que Scratch.

Le langage JavaScript sera utilisé dans le cadre du thème « web ».

Le langage Python sera utilisé pour l'étude de tous les autres thèmes, notamment lorsqu'il sera utile d'illustrer d'autres thèmes à l'aide de programmes.

Aspect expérimental du plan d'études

La nouveauté de la branche informatique a pour conséquence que le plan d'études a un caractère expérimental. Il conviendra donc d'évaluer dans 2 à 3 ans son adéquation à la pratique et de l'adapter le cas échéant.

² Les demi-classes sont dénotées par les lettres A et B.