

La discipline obligatoire *Informatique*

Proposition d'objectifs de formation pour les écoles de maturité

10 juillet 2018

Préambule

Le présent document propose un découpage thématique de la discipline obligatoire *Informatique* pour les écoles de maturité. Il a vocation à fournir une base de réflexion lors de l'élaboration de plans d'études cantonaux. Il propose une liste d'objectifs de formation répartis selon quatre axes:

1. Algorithmique et programmation, 2. Information et données, 3. Systèmes numériques et réseaux et 4. Informatique et société. Il positionne l'enseignement de l'informatique comme visant à l'acquisition, d'une part, des méthodes de pensée et d'action, et, d'autre part, un ensemble de notions et d'outils permettant de modéliser et de résoudre divers problèmes de façon computationnelle — sans se limiter à des problèmes issus de la discipline Informatique.

La répartition des objectifs de formation ne doit pas être vue comme une ligne directrice pour un découpage thématique strict des années de formation sur lesquelles l'enseignement de l'informatique sera dispensé. Une fois les bases respectives acquises, l'enseignement de l'informatique pourra favorablement s'enrichir de thématiques qui mêlent plusieurs de ces catégories. Par exemple, des concepts de robotique peuvent être vus comme étant à l'intersection des axes 1 et 3; la compression de données, à l'intersection des axes 1 et 2; ou la cryptographie, à l'intersection de tous les axes. En particulier, l'acquisition des composantes scientifiques et techniques des trois premiers axes permettra d'aboutir à une compréhension et à une critique éclairées de la dimension sociétale du quatrième axe.

Lorsque jugée pertinente, une approche par projet permettra de mettre en relation des savoirs et des savoir-faire rattachés à des objectifs de formation distincts.

Lignes directrices de la formation

Le cours d'informatique en tant que discipline obligatoire donnera aux élèves l'occasion de développer:

- des compétences de modélisation et programmation pouvant être mises en œuvre dans un contexte pluridisciplinaire;
- une approche de résolution de problèmes formelle et systématique en décomposant une tâche complexe en sous-tâches plus simples;
- une créativité technique et la pratique de l'autoévaluation pour la résolution de problèmes concrets;
- de nouvelles compétences informatiques de manière autonome.

Objectifs de formation

Les objectifs de formations sont répartis en quatre axes thématiques principaux. Ils se lisent en préfixant l'intitulé par «*au terme du cours, les élèves doivent être capables d'/de*».

1. Algorithmique et programmation

Si l'algorithmique et la programmation ne résument pas à elles seules l'informatique, elles n'en restent pas moins les outils centraux. Les buts principaux sont l'abord structuré de problèmes, leur décomposition, la formulation d'une liste d'étapes visant à les résoudre de façon systématique et l'implémentation dans un langage de programmation. Pour les atteindre, un travail d'alphabétisation et de familiarisation à l'algorithmique est nécessaire.

1.1 Identifier et décrire les principes de base de la logique algorithmique

Avant d'élaborer ou d'analyser des algorithmes complets, il faut connaître les briques élémentaires qui les composent. Ceci inclut notamment les variables, les conditions, les boucles et les sous-programmes. Il s'agit d'une alphabétisation de base indispensable, sur laquelle reposent les activités d'algorithmique et de programmation.

1.2 Lire le code d'un programme et en décrire les effets

Pour être capable de décrire les effets d'un programme, il faut se construire une représentation précise de la manière dont ce programme est exécuté. La lecture et la compréhension de code aident directement à développer des compétences de production de code.

1.3 Expliquer et implémenter quelques algorithmes classiques

À l'instar de la lecture des grands auteurs de la littérature, indispensable à l'étude d'une langue, la compréhension d'algorithmes importants permet d'acquérir une culture algorithmique de base. Les élèves sont confrontés à des algorithmes classiques de la discipline (tels que des algorithmes de tri ou de recherche) dont la connaissance est bénéfique par les principes qu'ils mettent en œuvre (approche «diviser pour régner», récursivité) — sans pour autant s'attendre à ce que les élèves puissent les élaborer ex nihilo.

1.4 Concevoir, écrire, documenter et évaluer des programmes

Il s'agit ici de la production de programmes de bout en bout. Celle-ci passe par l'analyse d'un problème à résoudre, la réflexion algorithmique, l'implémentation dans un langage donné, la documentation du code produit, le débogage et l'évaluation du programme final en termes d'exactitude et d'efficacité. Cette approche est menée sur des problèmes de difficulté raisonnable en termes de notions mathématiques et algorithmiques à mettre en œuvre dans leur résolution.

1.5 Manipuler les structures de données adaptées à un problème

La conception d'algorithmes est indissociable de la mise en œuvre des structures de données appropriées aux problèmes traités. Selon les problèmes, il peut s'agir, entre autres, de structures linéaires (p. ex. tableaux, listes chaînées, piles), hiérarchiques (arbres) ou associatives (dictionnaires). Sans entrer dans les tous les détails d'implémentation ou des propriétés théoriques, les grandes lignes permettant l'utilisation concrète de plusieurs structures courantes sont abordées.

2. Information et données

L'informatique est souvent définie comme la science du traitement automatique de l'information. Au cœur de cette définition se trouve la notion d'information, comprise au sens large, à la fois sous la forme de nombres, de textes ou d'images. L'ordinateur stocke des données, sans leur attribuer de sens. Une fois interprétées, ces données deviennent de l'information.

2.1 Expliquer les différentes représentations numériques de l'information et mentionner leurs spécificités

Les données peuvent être stockées sous différentes formes dans un ordinateur, mais en fin de compte tout n'est que 1 et 0. Ainsi, le même code binaire peut représenter une fois un nombre grâce au complément à deux, une autre fois un caractère grâce à la norme ASCII, voire une couleur grâce au code RGB. Toute information traitée par un ordinateur, qu'il s'agisse d'un programme, des événements de la souris ou d'un logiciel affiché à l'écran, est réduite à une information numérique: des 0 et des 1.

2.2 Évaluer les caractéristiques de différents systèmes de stockage de données

Afin de pouvoir les traiter, les données doivent être stockées en mémoire. Elles peuvent être stockées dans différentes structures de données adaptées au type de l'information, énumérées dans le point 1.5 ci-dessus. À plus grande échelle, les données peuvent être stockées séquentiellement dans un fichier faisant partie d'un système d'organisation hiérarchique. À très grande échelle, les données volumineuses sont stockées dans une base de données, offrant une recherche efficace et des garanties d'intégrité et de pérennité.

2.3 Analyser des données et interpréter les résultats

Toutes sortes de données peuvent être représentées numériquement. Il est possible d'en extraire la structure afin de répondre à des questions importantes: résumer et visualiser les données, faire des prévisions ou tirer des conclusions en vue d'une décision. En s'appuyant sur des exemples tirés de l'actualité ou d'autres disciplines, l'élève saisira la puissance de cette approche, mais également sa limite: la qualité des résultats égalera au mieux la qualité des données.

2.4 Résoudre un problème par modélisation et simulation

L'informatique a révolutionné la résolution de problèmes et ainsi transformé le visage de nombreuses sciences. Bien des problèmes complexes ne peuvent pas être résolus avec des méthodes analytiques, mais peuvent l'être numériquement. Par exemple, un phénomène d'origine physique, biologique, économique (ou autre) peut être représenté avec un modèle simplifié. En simulant ce phénomène, c'est-à-dire en imitant son comportement au cours du temps, il est possible d'en prédire l'évolution. En modélisant et simulant un problème simple provenant d'une autre discipline, l'élève se prépare à mieux comprendre la nature et l'incertitude d'une grande partie des découvertes scientifiques à venir.

3. Systèmes numériques et réseaux

Les systèmes numériques sont construits à partir de briques de base logiques et matérielles: des transistors sont assemblés pour former des composants, partie intégrante des ordinateurs, qui ne sont eux-mêmes que des points de réseaux plus vastes. Leur connaissance est la base nécessaire pour réfléchir à des problématiques informatiques actuelles ou futures.

3.1 Décrire les composants et le fonctionnement d'un ordinateur

Le but d'un ordinateur est d'exécuter des programmes sur des données. Son architecture fait appel à des éléments théoriques (machine de von Neumann, algèbre booléenne), matériels (transistor, porte logique, processeur, mémoire...) et logiciels (langage machine, assembleur, langages plus proches de notre langage naturel). Ces principes architecturaux sont communs à la plupart des équipements informatiques et permettent d'en comprendre les limites et les possibilités.

3.2 Expliquer l'organisation d'un réseau informatique et l'utilité de protocoles de communication

Les réseaux informatiques permettent aux ordinateurs de communiquer entre eux. Ils offrent de nombreux services tels que le web, la messagerie, l'échange de fichiers ou la téléphonie. Pour échanger des informations, les ordinateurs suivent des règles très précises définies dans des protocoles. Même s'ils sont nombreux et interagissent de manière complexe, ils suivent des principes communs. La connaissance des concepts fondamentaux des réseaux (adressage, routage, détection et correction des erreurs...) permet d'en comprendre l'utilité, le fonctionnement et la mise en œuvre par les ordinateurs et les équipements spécialisés (modem, routeurs...)

3.3 Décrire le principe de couches d'abstraction dans un contexte donné

La notion abstraite de couche est souvent utilisée en informatique pour construire des systèmes complexes qui dépendent les uns des autres. Une couche regroupe les fonctionnalités des couches inférieures pour offrir de nouveaux services aux couches supérieures, comme la couche matérielle qui offre des services à la couche logicielle ou le système d'exploitation qui offre des services aux programmes de l'utilisateur. La notion de couche est particulièrement importante pour décrire les réseaux de communication (modèle ISO et pile TCP/IP), dans lesquels chaque couche repose sur des protocoles bien précis.

3.4 Expliquer les principes de la sécurité de l'information

La sécurité de l'information a des fondements théoriques et pratiques. Les principes théoriques (disponibilité, intégrité, confidentialité...) permettent d'analyser les situations. Les aspects pratiques tels que la connaissance des types d'attaques, des contre-mesures et des bonnes pratiques permettent de prévenir et gérer les risques. La gestion de la sécurité de l'information ne s'arrête pas aux considérations techniques, mais s'inscrit aussi dans des contextes légaux et humains.

4. Informatique et société

Les élèves doivent être sensibilisés aux aspects sociétaux plus larges de l'informatique pour développer une compréhension des questions culturelles, sociales, éthiques et légales qui y sont associées. Il est également opportun de présenter une perspective historique de la discipline, en abordant notamment les tendances de fond et les contributions de quelques-uns des pionniers du domaine.

4.1 Distinguer les enjeux éthiques, légaux et sociétaux du monde de l'information

Alors que les avancées technologiques bouleversent chaque jour des parts plus importantes du monde qui nous entoure, de nouvelles problématiques sociales émergent. Les élèves développent leur capacité à identifier les enjeux éthiques, légaux ou sociétaux inhérents à des questions contemporaines dans lesquelles l'informatique joue un rôle majeur (p. ex. démocratie électronique, collecte massive de données, cybercriminalité). Ils apprennent à

identifier les défis posés aux individus et aux sociétés par les développements les plus significatifs de l'informatique dans le monde contemporain, notamment l'apprentissage automatique et l'intelligence artificielle.

4.2 Évaluer les apports et limites de l'informatique et de ses développements dans le cadre de problématiques sociétales

En s'appuyant sur des connaissances techniques et scientifiques, les élèves formulent des questions pertinentes au sujet du rôle et des impacts de l'informatique dans ces différentes situations et évaluent les réponses qui y sont proposées. Ils développent leur esprit critique et font la part des choses entre les avantages, les limites, mais également les dérives que peut amener une solution technologique dans une situation donnée.

4.3 Produire un discours susceptible de contribuer à la problématisation publique d'enjeux techniques complexes

Ayant identifié des situations comportant des enjeux sociétaux et évalué à l'intérieur de celles-ci les apports et limites de l'informatique, les élèves sont à même de s'exprimer sur l'une de ces problématiques auprès d'un public non spécialiste. Ils synthétisent les impacts de l'informatique sur ces situations, présentent les détails techniques de manière simplifiée tout en montrant les aspects complexes des questions posées. Ils peuvent ainsi formuler un avis personnel instruit et, de là, exercer une citoyenneté active et responsable.

4.4 Situer et mettre en perspective les principaux repères historiques de l'informatique

L'histoire de l'informatique est enseignée pour faire ressortir la manière dont l'informatique et les technologies ont eu, dans notre passé récent, un impact sur la société à une très large échelle. Les ordinateurs et internet ont, peut-être plus que toute autre technologie, transformé la société depuis le milieu du xx^e siècle. Comme dans toute approche historique, la connaissance du passé nous apporte une meilleure compréhension du présent et nous ouvre des portes vers un avenir souhaitable.

Contributeurs

Ont contribué à la rédaction de ce document (dans l'ordre alphabétique des noms de famille):

Karim Benmachiche	VD	Gabriel Parriaux	HEP VD
Nicolas Bique	FR	Jean-Philippe Pellet	HEP VD; EPFL
David Da Silva	VD	Biljana Petreska von Ritter	VD
Javier Iglesias	VD	Philippe RoCHAT	VD
Murièle Jacquier	GE	Fabian Simillion	FR
Dominique-Alain Jan	VD	Gabriel Thullen	GE
Arnaud Le Gourriérec	HEP BEJUNE	Pierre-Olivier Vallat	HEP BEJUNE
Michaël Lew	OLPC	Samuel Vannay	VS
Alexandre Lopes	VD		
Isaac Pante	UNIL		

Contact

Jean-Philippe Pellet, jean-philippe.pellet@hepl.ch
Samuel Vannay, samuel.vannay@eduvs.ch