



**Département d'informatique**  
**IFT 209 – Programmation système**

**Plan de cours**  
**Automne 2016**

---

**Enseignant****Richard St-Denis**

---

Courriel :	<a href="mailto:Richard.St-Denis@USherbrooke.ca">Richard.St-Denis@USherbrooke.ca</a>
Local :	D4-2005
Téléphone :	(819) 821-8000 poste 62847
Site :	<a href="http://www.usherbrooke.ca/informatique/personnel/professeurs/professeurs/richard-st-denis/">http://www.usherbrooke.ca/informatique/personnel/professeurs/professeurs/richard-st-denis/</a>
Disponibilité :	horaire de consultation affiché sur ma porte

---

**Horaire**

Résumé et exercice en classe :	mardi	10 h 30 à 12 h 20	salle D3-2034
Laboratoires :	mercredi	8 h 30 à 10 h 20	salle D3-2034 ou D4-1017 ou D4-1023

---

**Description officielle de l'activité pédagogique <sup>1</sup>**

Cibles de formation	Comprendre l'architecture d'un ordinateur, les systèmes de numération, les types élémentaires de données, les structures de contrôle, les entrées-sorties; savoir effectuer une programmation-système.
Contenu	Introduction à l'architecture des ordinateurs. Système de numération. Modes d'adressage. Format des instructions machine. Représentation des données. Technique de mise au point de programmes. Arithmétique entière. Arithmétique à virgule flottante. Manipulation de bits. Sous-programmes. Application à une architecture contemporaine. Entrées-sorties. Traitement des interruptions.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel par semaine
Préalable	IFT 159
Particularités	Cette activité pédagogique se présente sous la forme de lectures et de résolution de problèmes par les étudiantes et les étudiants.

---

1. [www.usherbrooke.ca/fiches-cours/ift209](http://www.usherbrooke.ca/fiches-cours/ift209)

# 1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation du comité de programme du Département d'informatique, constitue la version officielle.

## 1.1 Mise en contexte

*À propos des langages d'assemblage . . .*

Lorsque les premiers ordinateurs sont apparus, vers la fin des années 1940, la programmation s'effectuait en langage machine, car à cette époque les outils de programmation étaient rudimentaires ou inexistantes. La programmation constituait une tâche très fastidieuse puisque toutes les instructions et les données devaient être codées manuellement à l'aide d'une suite de chiffres 0 et 1. La moindre erreur pouvait entraîner plusieurs jours de travail additionnels.

Dans le but de réduire le travail des programmeurs, les informaticiens de l'époque inventèrent des langages de programmation. Les premiers furent les langages d'assemblage qui existent encore de nos jours. Ils facilitent l'écriture de programmes en utilisant une notation symbolique dans laquelle les instructions machine, les données, ainsi que les adresses de la mémoire principale sont représentées par des noms symboliques. Des modifications peuvent désormais être apportées aux programmes sans devoir les récrire complètement, les suites de chiffres 0 et 1 étant remplacées par des noms symboliques. Peu à peu les langages d'assemblage ont fait place à des langages de plus en plus évolués qui ont contribué à une amélioration sensible de la production de programmes.

Même si aujourd'hui les programmeurs écrivent rarement des programmes en langage d'assemblage, il n'en demeure pas moins que ces derniers sont encore utiles. D'une part, ils servent principalement dans la construction de compilateurs ainsi que dans la mise en oeuvre de systèmes d'exploitation et de systèmes embarqués. D'autre part, ils sont un outil précieux dans l'introduction et l'expérimentation des notions de base en architecture des ordinateurs. Enfin, ils permettent de mieux saisir les concepts fondamentaux des langages de programmation.

*À propos des entrées/sorties et des interruptions . . .*

De tout temps, la lenteur des entrées/sorties par rapport à la vitesse du processeur a posé le problème de l'interaction entre le processeur et les périphériques d'entrées/sorties. Une solution adéquate à ce problème nécessite la compréhension des mécanismes matériels et logiciels de synchronisation, en particulier le mécanisme d'interruption qui permet la gestion d'évènements internes aux activités du processeur ou d'évènements externes au processeur. La compréhension du mécanisme d'une interruption est indispensable à l'étude des principes sous-jacents aux systèmes d'exploitation.

*À propos de la place de cette activité pédagogique dans votre programme . . .*

L'activité pédagogique intitulée *Programmation système* est une activité pédagogique de base. Elle est préalable aux activités pédagogiques obligatoires suivantes des programmes d'informatique et d'informatique de gestion :

- IFT 320 — Systèmes d'exploitation ;
- IFT 585 — Télématique.

Elle est également préalable, dans le programme d'informatique *sans concentration* et dans la concentration *systèmes et réseaux*, aux activités pédagogiques à option suivantes :

- GEI 201 — Circuits logiques ;
- GEI 301 — Architecture et organisation des ordinateurs.

Sans en être un préalable, elle est utile à l'activité pédagogique à option suivante du programme d'informatique :

- IFT 580 — Compilation et interprétation des langages.

## 1.2 Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable :

1. de maîtriser l'arithmétique dans plusieurs systèmes de numération ;
2. de connaître et d'expliquer des notions de base en architecture des ordinateurs ;
3. de connaître et d'expliquer des types élémentaires de données comme les entiers, les nombres en virgule flottante, les tableaux, les chaînes de bits et les chaînes de caractères ;
4. de connaître et d'expliquer des structures de contrôle comme la séquence, la sélection, l'itération et les sous-programmes ;
5. d'utiliser des méthodes élémentaires dans le développement de programmes ;
6. de comprendre les concepts élémentaires des langages procéduraux à travers ceux des langages d'assemblage ;
7. d'écrire de petits programmes (100 à 350 instructions) dans un langage d'assemblage ;
8. de connaître et d'expliquer les opérations de lecture et d'écriture sur un périphérique d'entrées/sorties ;
9. de comprendre le mécanisme d'interruption et les différentes étapes de sa gestion.

## 1.3 Contenu détaillé

Le tableau suivant contient la matière présentée dans ce cours. L'étude de cette matière est accompagnée d'au moins six devoirs couvrant tous les sujets, dont au moins quatre travaux de programmation.

Thème	Contenu	N <sup>bre</sup> d'heures	Objectifs	Références
1	Introduction : présentation du plan de cours ; présentation de la programmation en langage d'assemblage.	2		
2	Systèmes de numération : écriture des nombres dans un système de numération ; conversion de nombres d'un système de numération en un autre.	2	1	chap. 1 de [3]
3	Architecture des ordinateurs : architecture von Neumann ; mémoire principale, processeur, registres ; jeux d'instructions ; aspects particuliers à l'organisation d'un ordinateur ; processeur <i>UltraSPARC</i> .	4	2	chap. 2 de [3]
4	Accès aux données : données, adresses, modes d'adressage ; étapes de la vie d'un programme.	4	2	chap. 3 de [3]
5	Programmation en langage d'assemblage : survol à partir d'un petit programme ; différence entre une instruction et une pseudo-instruction ; programmation de haut niveau des entrées/sorties.	4	3, 4, 5, 7	chap. 4 de [3]
6	Les nombres entiers : représentations des entiers signés et non signés ; report et débordement ; instructions arithmétiques.	4	3, 6, 7	chap. 5 de [3]
7	Tableaux : tableaux à une dimension et à deux dimensions.	2	3, 6, 7	chap. 6 de [3]
8	Structures de contrôle : condition, branchements, séquence, sélection, itération ; appel et retour de sous-programmes ; notion d'environnement ; récursivité.	4	4, 6, 7	chap. 7 de [3]
9	Chaînes de bits : opérations logiques, décalages.	2	3, 6, 7	chap. 8 de [3]
10	Chaînes de caractères : codes de représentation de caractères ; opérations sur les caractères et les chaînes de caractères.	2	3, 6, 7	chap. 9 de [3]
11	Sous-programmes appel et retour, passage de paramètres, sauvegarde/récupération, récursivité.	4	4, 6, 7	chap. 10 de [3]
12	Les nombres en virgule flottante : représentations des nombres en virgule flottante ; erreur d'arrondi, erreur de troncation, dépassement de capacité ; norme IEEE 754 ; instructions arithmétiques.	4	3, 6, 7	chap. 11 de [3]
13	La programmation des entrées/sorties : interrogation de bits d'états, interruption, niveaux de priorités des interruptions ; illustration à l'aide de dispositifs simples.	10	8, 9	

## 2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

### 2.1 Utilisation d'appareils « intelligents » et courriels

1. L'utilisation de téléphones cellulaires est interdite en tout temps dans la salle de classe et les laboratoires.
2. L'utilisation de tablettes intelligentes ou d'ordinateurs est interdite en tout temps dans la salle de classe à moins d'avoir obtenu personnellement l'autorisation du professeur. Cette permission peut être retirée en tout temps, si l'appareil n'est pas utilisé uniquement à des fins d'apprentissage.
3. Je ne réponds à aucun courriel.

### 2.2 Méthode pédagogique

L'activité pédagogique *Programmation système* est avant tout orientée vers la compréhension du fonctionnement des ordinateurs à travers l'écriture de programmes en utilisant des langages de programmation proches de la machine, en particulier les langages d'assemblage. C'est pour cette raison qu'elle exige de la part des étudiantes et des étudiants une participation active. Basée sur le manuel de cours [3] qui couvre la totalité de la matière, à l'exception du dernier thème, une semaine typique se présente sous la forme suivante:

1. lecture par les étudiantes et les étudiants des parties du livre du cours portant sur la matière à assimiler;
2. présentation sommaire par le professeur des points les plus difficiles de la matière (environ une heure);
3. résolution de problèmes en classe par les étudiantes et les étudiants avec l'aide du professeur (environ une heure);
4. expérimentation des concepts sous la forme d'un laboratoire par les étudiantes et les étudiants (deux heures).

Toutes les notions théoriques relatives aux différents thèmes sont illustrées à l'aide d'exemples construits à partir de l'architecture *SPARC version 9* et de programmes en langage d'assemblage *SPARC version 9*. De plus, l'architecture *ARM version 7* est utilisée pour illustrer la programmation des entrées/sorties et des interruptions [1].

### 2.3 Calendrier

Semaines	Thèmes	Lectures	Laboratoires	Devoirs
1	1–2	notes de cours, chapitre 1 de [3]		
2	2–4	chapitres 2 et 3 de [3]	laboratoire #1	
3	3–5	chapitre 4 de [3]	laboratoire #2	
4	12	chapitre 11 de [3]	laboratoire #3	
5	5–6	chapitre 5 de [3]	laboratoire #4	
6	4–7	chapitre 6 de [3]	exercice dirigé	
7	examen périodique			
8	8	chapitre 7 de [3]		devoir #1
9	4–10	chapitres 8 et 9 de [3]	laboratoire #5	
10	4–10, 13	notes de cours	laboratoire #6	devoir #2
11	13	notes de cours	laboratoire #7	
12	11, 13	chapitre 10 de [3]	laboratoire #8	
13	11	chapitre 10 de [3]	exercice dirigé	
14	12	chapitre 11 de [3]	exercice dirigé	
15–16	examen final			

## 2.4 Évaluation

Laboratoires	20%
Devoirs	20%
Examen périodique	30%
Examen final	30%
Qualité du français	0%

Toute documentation est permise aux examens. Toutefois, les appareils électroniques (baladeur numérique, téléphone cellulaire, calculatrice, portable, etc.) sont interdits pendant les examens.

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages des étudiantes et des étudiants

(<http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-d-evaluation/>),

l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

Toute situation de plagiat sera traitée en conformité, entre autres, avec l'article 8.1.2 du *Règlement des études* de l'Université de Sherbrooke disponible à l'adresse

<http://www.usherbrooke.ca/programmes/references/reglement>

Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 8.1.2. En particulier,

Un document dont le texte et la structure se rapporte à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat, défini dans le *Règlement des études* comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'oeuvre d'autrui ». Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du *Règlement des études* : « tout acte ou manoeuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique. » À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera référé au Secrétaire de la Faculté des sciences.

## 2.5 Échéancier des travaux

Les directives, la date de remise et le barème relatifs aux laboratoires et aux devoirs seront connus à la remise de l'énoncé de chaque d'eux aux étudiantes et aux étudiants.

**Directives particulières :** Un laboratoire a lieu chaque semaine afin que les étudiantes et les étudiants assimilent les concepts du cours. Aussi, deux devoirs qui comportent l'écriture de programmes sont donnés aux étudiantes et aux étudiants pour qu'ils approfondissent la programmation en langage d'assemblage *SPARC* ou la programmation d'un module d'un micro-processeur *ARM* en langage de programmation *C*. **Les laboratoires et les devoirs peuvent être faits individuellement ou par équipe de deux personnes. Aucun laboratoire ou devoir ne peut être remis par courrier électronique. Les laboratoires et devoirs non remis reçoivent automatiquement la note zéro.**

La correction des devoirs et des examens est entre autres basée sur le fait que chacune de vos réponses soit :

- claire, c'est-à-dire lisible et compréhensible pour le correcteur ;
- précise, c'est-à-dire exacte ou sans erreur ;
- complète, c'est-à-dire que toutes les étapes de résolution du problème sont présentes ;
- concise, c'est-à-dire que la méthode de résolution est la plus courte possible.

### 3 Matériel nécessaire pour le cours

Les notes de cours [3] sont obligatoires. L'architecture *SPARC version 9* est entièrement décrite dans la référence [2]. Tous les programmes et les diapositives présentés dans le cours sont disponibles dans le répertoire `Public/Cours` dont l'accès est décrit dans la page Web

<http://www.usherbrooke.ca/informatique/intranet/ressources-et-documentation/faq/acces-aux-lecteurs-reseaux>

Les programmes des notes de cours [3] écrits dans la version 9 de l'architecture *SPARC* sont disponibles dans le répertoire public du cours.

### 4 Références

- [1] Outils de développement pour modules ARM.  
<http://www.siwawi.arubi.uni-kl.de/avr-projects/arm-projects/index.html>.
- [2] SPARC International. *The SPARC Architecture Manual, version 9*. Prentice-Hall, 1994. Manuel disponible à la bibliothèque.
- [3] R. St-Denis. *L'architecture du processeur SPARC et sa programmation en langage d'assemblage*. 2015. Notes de cours disponibles à la Coop de l'Université de Sherbrooke.

---

## L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille!

---

### Extrait du Règlement des études

#### 8.1.2 Relativement aux activités pédagogiques

L'expression délit désigne d'abord tout acte ou toute manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique.

Sans restreindre la portée générale de ce qui précède, est considéré comme un délit :

- a) la substitution de personnes ou l'usurpation d'identité lors d'une activité évaluée ou obligatoire;
- b) le plagiat, soit le fait, dans une activité évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui;
- c) l'obtention par vol ou par toute autre manœuvre frauduleuse de document ou de matériel, la possession ou l'utilisation de tout matériel non autorisé avant ou pendant un examen ou un travail faisant l'objet d'une évaluation;
- d) le fait de fournir ou d'obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour un examen ou un travail faisant l'objet d'une évaluation;
- e) le fait de soumettre, sans autorisation préalable, une même production comme travail à une deuxième activité pédagogique;
- f) la falsification d'un document aux fins d'obtenir une évaluation supérieure dans une activité ou pour l'admission à un programme.

### Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets
- Reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire
- Utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources
- Résumer ou paraphraser l'idée d'un auteur sans en indiquer la source
- Traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets
- Utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord)
- Acheter un travail sur le Web ou ailleurs et le faire passer pour sien
- Utiliser sans autorisation le même travail pour deux activités différentes (autoplégat)

---

## Autrement dit : mentionnez vos sources.

---