

# Cours PROG : Introduction à la programmation L3 informatique, parcours Recherche et innovation Examen final, 12 décembre 2016

Programmation fonctionnelle Ocaml : Luc Bougé

Durée : 1 h 20 + 10 mn de relecture

Cette épreuve écrite a pour objectif de faire la synthèse de l'ensemble des sujets étudiés en cours. La rapidité n'est pas un critère majeur d'évaluation. Au contraire, l'accent est mis sur le soin, la précision, la pédagogie, l'inventivité, la capacité à bien exploiter l'ensemble des documents que vous avez apportés, etc. Vous êtes invités à utiliser l'ensemble des documents disponibles : notes de cours, photocopiés, livres, etc. Seuls les PC et Internet ne sont pas les bienvenus.

La forme de l'épreuve est inspirée de celle de l'épreuve de leçon à l'Agrégation. Vous recevez un sujet et vous imaginez que vous avez à faire un cours de 2 heures à une classe de L3 sur ce sujet. Que proposeriez-vous ? Comme pour la préparation d'un exposé, un temps de relecture est prévu pour la correction de l'orthographe et le soin de la présentation. Ces aspects seront notés sur 20% du barème. Il est attendu deux productions.

**Un plan de cours**, qui devrait tenir sur 1 page, 2 éventuellement, sûrement pas plus. Ce plan n'est pas une dissertation, c'est plutôt une liste structurée de points à aborder dans ce cours, avec quelques mots pour chacun des points.

Vous pouvez imaginer par exemple que c'est ce que vous auriez préparé sur vos notes de cours, ou ce que vous écririez au tableau ou sur vos diapos. Pas de longues phrases, mais un style télégraphique comme dans les diapos de soutenance.

**Un point développé**, choisi par vous, pour lequel vous rédigez soigneusement ce que vous proposez de dire aux étudiants. La longueur recommandée est 2 pages, 3 éventuellement, sûrement pas plus.

## **Critères d'évaluation des productions**

### **Plan : 1–2 pages, pas plus**

**Complétude** de la couverture des aspects traités

**Organisation** de l'exposé, graduation, équilibre

**Précision** de la rédaction, soin, orthographe

**Qualité** des exemples, pédagogie, originalité, inventivité

### **Développement : 2–3 pages, pas plus**

**Intérêt** du point choisi pour le développement

**Organisation** du développement, pédagogie

**Précision** de la rédaction, soin, orthographe

**Qualité** des exemples, pédagogie, originalité, inventivité

## Sujets proposés au choix

Il vous est demandé de choisir l'un des deux sujets et d'y consacrer toute votre attention. Si vous traitez plus d'un sujet, seul le premier sera corrigé.

### 1 Types

CamL est un langage fortement typé avec synthèse de type complète. Préparez une leçon pour expliquer cette approche, ses avantages et ses inconvénients, à la fois sur le plan pratique de la programmation et aussi sur le plan théorique de la sémantique du langage.

#### suggestions

Montrez comment les types peuvent être construits, notamment pour l'ordre supérieur, et comment ils sont utilisés pour les différentes constructions (filtrage, exceptions, etc.)

Présentez les contraintes de typage définies par le langage sur les structures de contrôle : ; (séquence), **if** (choix), **for** (itération bornée), **while** (itération non bornée), **try** (exception), etc.

Discutez des avantages et des inconvénients pour CamL du choix d'un typage avec synthèse complète, complètement automatique.

Montrez comment l'utilisation des types fonctionnels d'ordre supérieur (*continuations*) permet de rendre compte de notions avancées de programmation (exceptions, échappements, etc.)

Bien sûr, ces suggestions sont indicatives. Toute idée originale, si elle est bien présentée et justifiée, sera hautement appréciée. De même, la qualité des exemples mentionnés sera attentivement appréciée.

### 2 Références

CamL permet de manipuler des objets modifiables grâce au concept de *référence*. Ce concept a donc une importance centrale dans le langage. Préparez une leçon pour présenter ce concept, son utilisation et son interaction avec les autres aspects du langage.

#### suggestions

Introduisez la notion de référence, présentez quelques exemples simples, montrez comment une référence est typée.

Montrez que d'autres objets modifiables du langage peuvent se ramener à des références : tableaux, enregistrements modifiables (*mutable*), etc.

Détaillez les interactions entre les références et le typage. Montrez en particulier comment l'utilisation des références peut conduire des objets avec des types partiellement définis ('\_a, etc.), dont le type est ensuite fixé à l'exécution.

Les références sont à la base de la notion d'*effet de bord*. Présentez des exemples typiques d'effets de bord. Montrez qu'ils peuvent introduire une indétermination du résultat d'un programme.

Bien sûr, ces suggestions sont indicatives. Toute idée originale, si elle est bien présentée et justifiée, sera hautement appréciée. De même, la qualité des exemples mentionnés sera attentivement appréciée.