

TYLA – Typologie des langages

EPITA – AppIng1 promotion 2014 – **Sans documents ni machine**

Juin 2012 (1h30)

Bien lire les questions, chaque mot est important. Écrire court, juste, et bien. Une argumentation informelle mais convaincante est souvent suffisante.

Répondre aux questions sur les formulaires de QCM ; aucune réponse manuscrite ne sera corrigée. Ne pas oublier de renseigner les champs d'identité. Il y a exactement une et une seule réponse juste pour ces questions. Si plusieurs réponses sont valides, sélectionner la plus restrictive. Par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négalif*, cocher *nul* qui est plus restrictif que *positif* et *négalif*, tous deux vrais. Répondre incorrectement à une question à choix multiples est plus pénalisé que de ne pas répondre.

Ce sujet contient 5 pages. Les pages 1-4 contiennent l'épreuve elle-même. Ce document comporte en page 5 une enquête (facultative) sur le cours ; y répondre sur la feuille de QCM.

1 Langages

- Q.1 Quel langage ne repose pas sur une machine virtuelle pour assurer l'exécution du code ?
- | | | | |
|-------|-------|---------|----------------|
| a. C# | b. Go | c. Perl | d. Pascal UCSD |
|-------|-------|---------|----------------|
- Q.2 Lequel de ces langages est typé statiquement ?
- | | | | |
|---------------|-----------|----------|---------|
| a. Javascript | b. MATLAB | c. OCaml | d. Ruby |
|---------------|-----------|----------|---------|
- Q.3 À quel nom ne peut-on associer John Backus ?
- | | | | |
|----------|--------|----------|------------|
| a. ALGOL | b. BNF | c. COBOL | d. FORTRAN |
|----------|--------|----------|------------|

2 Fonctions

- Q.4 Le support des fonctions récursives nécessite
- | | |
|-------------------------|--------------------------------------|
| a. une pile. | c. un <i>static link</i> . |
| b. des types fonctions. | d. des <i>forward declarations</i> . |
- Q.5 Les variables non locales sont permises grâce à
- | | |
|--------------------|-----------------------------|
| a. <i>auto</i> . | c. la structure de blocs. |
| b. <i>static</i> . | d. la déclaration en blocs. |
- Q.6 En C, la ligne suivante

```
int foo = getc () + getc () * getc () ;
```

- | | |
|--------------------------------------|--|
| a. est invalide. | d. dépend de la plate-forme et/ou de l'implémentation. |
| b. n'est pas déterministe. | |
| c. produit le même résultat partout. | |
- Q.7 À la fin de ce programme, avec un *Mode* de passage des arguments par valeur (copie), quelles sont les valeurs des l-values ?

```

var t      : integer
    foo   : array [1..2] of integer;

procedure shoot_my(x : Mode integer);
begin
    foo[1] := 5;
    t      := 1;
    x      := x - 1;
end;

```

```

begin
    foo[1] := 0;
    foo[2] := 1;
    t      := 2;
    shoot_my (foo[t]);
end.

```

- a. foo[1] = 0, foo[2] = 1, t = 1
- b. foo[1] = 4, foo[2] = 1, t = 1
- c. foo[1] = 5, foo[2] = 0, t = 1
- d. foo[1] = 5, foo[2] = 1, t = 1

Q.8 Même question, mais avec un *Mode* de passage d'arguments par valeur-résultat, à la Algol W. On rappelle qu'en Algol W, la l-value dans laquelle est copiée la valeur d'un argument passé par résultat ('out') est évaluée *au retour* de la fonction.

- a. foo[1] = 0, foo[2] = 1, t = 1
- b. foo[1] = 4, foo[2] = 1, t = 1
- c. foo[1] = 5, foo[2] = 0, t = 1
- d. foo[1] = 5, foo[2] = 1, t = 1

Q.9 Même question, mais avec un *Mode* de passage d'arguments par valeur-résultat, à la Ada. On rappelle qu'en Ada, la l-value dans laquelle est copiée la valeur d'un argument passé par résultat ('out') est évaluée *à l'appel* de la fonction.

- a. foo[1] = 0, foo[2] = 1, t = 1
- b. foo[1] = 4, foo[2] = 1, t = 1
- c. foo[1] = 5, foo[2] = 0, t = 1
- d. foo[1] = 5, foo[2] = 1, t = 1

Q.10 Même question, mais avec un *Mode* de passage d'arguments par référence.

- a. foo[1] = 0, foo[2] = 1, t = 1
- b. foo[1] = 4, foo[2] = 1, t = 1
- c. foo[1] = 5, foo[2] = 0, t = 1
- d. foo[1] = 5, foo[2] = 1, t = 1

Q.11 Même question, mais avec un *Mode* de passage d'arguments par nom.

- a. foo[1] = 0, foo[2] = 1, t = 1
- b. foo[1] = 4, foo[2] = 1, t = 1
- c. foo[1] = 5, foo[2] = 0, t = 1
- d. foo[1] = 5, foo[2] = 1, t = 1

Q.12 Comment peut-on simuler le passage par nom en C++ ?

- a. Avec des macros.
- b. Avec des concepts.
- c. En utilisant void*.
- d. Il n'y a aucun moyen de simuler du passage par nom en C++.

Q.13 La liaison retardée en C++

- a. repose sur `template`.
- b. s'appuie sur `dynamic_cast`.
- c. a un rapport avec `virtual`.
- d. est liée à la surcharge des opérateurs.

- b. `std::pair<float, int> p2 (42, 3.14f);`
- c. `std::pair<int, float> p3 = std::make_pair (42, 3.14f);`
- d. `std::pair<int, float> p4 = std::make_pair<int, float> (2.72f, 51);`

Q.25 Le code C++ ci-dessous :

```
#include <complex>
#include <vector>
#include <algorithm>

int
main()
{
    std::vector< std::complex<float> > v;
    v.push_back (std::complex<float>(1.f, 1.f));
    v.push_back (std::complex<float>(42.f, 51.f));
    v.push_back (std::complex<float>(4.3f, -2.2f));
    std::sort (v.begin(), v.end());
}
```

- a. compile correctement et s'exécute sans erreur.
- b. compile correctement mais provoque une erreur à l'exécution.
- c. provoque une erreur de compilation mentionnant l'absence d'un 'operator<'.
'
- d. provoque une erreur de compilation mentionnant explicitement l'inadéquation entre `std::complex<float>::iterator` et le concept `RandomAccessIterator`.

6 À propos de ce cours

Pour terminer cette épreuve, nous vous invitons à répondre à un petit questionnaire. Les renseignements ci-dessous ne seront bien entendu pas utilisés pour noter votre copie. Ils ne sont pas anonymes, car nous souhaitons pouvoir confronter réponses et notes. En échange, quelques points seront attribués pour avoir répondu. Merci d'avance.

Sauf indication contraire, vous pouvez cocher plusieurs réponses par question. Répondez sur la feuille de QCM. N'y passez pas plus de dix minutes.

Cette épreuve

Q.26 Sans compter le temps mis pour remplir ce questionnaire, combien de temps ce partiel vous a-t-il demandé (si vous avez terminé dans les temps), ou combien vous aurait-il demandé (si vous aviez eu un peu plus de temps pour terminer) ?

- | | |
|----------------------------|-----------------------------|
| a. Moins de 30 minutes. | d. Entre 90 et 120 minutes. |
| b. Entre 30 et 60 minutes. | e. Plus de 120 minutes. |
| c. Entre 60 et 90 minutes. | |

Q.27 Ce partiel vous a paru

- | | | |
|---------------------|------------------------------|------------------|
| a. Trop difficile. | c. D'une difficulté normale. | d. Assez facile. |
| b. Assez difficile. | | e. Trop facile. |

Le cours

Q.28 Quelle a été votre implication dans les cours TYLA ?

- | | |
|---------------------------------------|-------------------------|
| a. Rien. | d. Fait les annales. |
| b. Bachotage récent. | e. Lu d'autres sources. |
| c. Relu les notes entre chaque cours. | |

Q.29 Ce cours

- | | |
|---|--|
| a. Est incompréhensible et j'ai rapidement abandonné. | c. Est facile à suivre une fois qu'on a compris le truc. |
| b. Est difficile à suivre mais j'essaie. | d. Est trop élémentaire. |

Q.30 Ce cours

- | | |
|---|---|
| a. Ne m'a donné aucune satisfaction. | d. Est nécessaire mais pas intéressant. |
| b. N'a aucun intérêt dans ma formation. | e. Je le recommande. |
| c. Est une agréable curiosité. | |

Q.31 La charge générale du cours en sus de la présence en amphi (relecture de notes, compréhension, recherches supplémentaires, etc.) est

- | | |
|--|---|
| a. Telle que je n'ai pas pu suivre du tout. | c. Supportable (environ une heure par semaine). |
| b. Lourde (plusieurs heures de travail par semaine). | d. Légère (quelques minutes par semaine). |

Les formateurs

Q.32 L'enseignant

- | | |
|--|--|
| a. N'est pas pédagogue. | d. Se répète vraiment trop. |
| b. Parle à des étudiants qui sont au dessus de mon niveau. | e. Se contente de trop simple et devrait pousser le niveau vers le haut. |
| c. Me parle. | |