

[Accueil](#) / [Mes cours](#) / [2026 ING1 TYLA](#) / [Sections](#) / [Exam 15 Avril](#) / [TYLA 15 Avril](#)

Commencé le Monday 15 April 2024, 10:00

État Terminé

Terminé le Monday 15 April 2024, 10:11

Temps mis 10 min 55 s

Points 28,08/32,00

Note 17,55 sur 20,00 (87,76%)

Question **1**

Correct

Note de 1,00 sur 1,00

Quel langage a été le premier à avoir un compilateur optimisant ?

- a. COBOL
- b. ALGOL 60
- c. FORTRAN ✓
- d. C

Votre réponse est correcte.

La réponse correcte est :

FORTRAN

Question **2**

Partiellement correct

Note de 0,75 sur 1,00

Lesquelles de ces innovations ont été présentées en 1968 lors de la Mother of All Demos ?

- a. Les hyperliens ✓
- b. Le copier/coller
- c. La souris ✓
- d. Le transistor
- e. Les cartes perforées
- f. La télé-conférence ✓

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 3.

Les réponses correctes sont :

La souris,

La télé-conférence,

Le copier/coller,

Les hyperliens

Question 3

Correct

Note de 1,00 sur 1,00

Qui n'a pas participé au développement d'ALGOL 60 ?

- a. John McCarthy
- b. Peter Naur
- c. Niklaus Wirth ✓
- d. John Backus

Votre réponse est correcte.

La réponse correcte est :

Niklaus Wirth

Question 4

Correct

Note de 1,00 sur 1,00

A qui doit-on l'invention originelle d'Unix ?

- a. Ken Thompson ✓
- b. Bjarne Stroustrup
- c. Brian Kernighan
- d. Richard Stallman

Votre réponse est correcte.

La réponse correcte est :

Ken Thompson

Question 5

Correct

Note de 1,00 sur 1,00

Que signifie APL ?

- a. A Programming Language ✓
- b. Array Processing Language
- c. Abstract Programming Language
- d. Array Programming Language

Votre réponse est correcte.

La réponse correcte est :

A Programming Language

Question 6

Correct

Note de 1,00 sur 1,00

Quelle est la différence entre une fonction et une procédure ?

- a. Une procédure a une valeur de retour, une fonction n'en a pas.
- b. Une procédure peut effectuer des effets de bords, une fonction ne peut pas.
- c. Une fonction a une valeur de retour, une procédure n'en a pas. ✓
- d. Il n'y a pas de différence.
- e. Une fonction ne prend pas d'arguments, une procédure peut en prendre.

Votre réponse est correcte.

La réponse correcte est :

Une fonction a une valeur de retour, une procédure n'en a pas.

Question 7

Incorrect

Note de 0,00 sur 1,00

Lequel de ces mécanismes n'est pas une forme de généricité par boxing ?

- a. Pointeurs void* en C ✗
- b. Type-erasure
- c. Passage de dictionnaires
- d. Polymorphisation

Votre réponse est incorrecte.

La réponse correcte est :

Polymorphisation

Question 8

Correct

Note de 1,00 sur 1,00

La monomorphisation est un processus compile-time uniquement.

Veuillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question **9**

Correct

Note de 1,00 sur 1,00

La représentation uniforme d'OCaml est une forme de généricité par boxing.

Veillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question **10**

Correct

Note de 1,00 sur 1,00

L'instantiation des génériques est optionnelle en Ada.

Veillez choisir une réponse.

- Vrai
- Faux ✓

La réponse correcte est « Faux ».

Question **11**

Correct

Note de 1,00 sur 1,00

Les macros Rust sont un exemple de métaprogrammation

Veillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question **12**

Correct

Note de 1,00 sur 1,00

Les templates C++ sont compilés par monomorphisation

Veillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question **13**

Correct

Note de 5,00 sur 5,00

Associer l'auteur à son langage

Barbara Liskov	CLU	✓
Niklaus Wirth	Pascal	✓
Dennis Ritchie	C	✓
John Backus	FORTRAN	✓

Votre réponse est correcte.

La réponse correcte est :

Barbara Liskov → CLU,

Niklaus Wirth → Pascal, Dennis Ritchie → C,

John Backus → FORTRAN

Question **14**

Correct

Note de 5,00 sur 5,00

Associer les langages à leur date de parution

1956	Fortran	✓
1958	Algol	✓
1959	COBOL	✓
1972	C	✓
1996	OCaml	✓

Votre réponse est correcte.

La réponse correcte est :

1956 → Fortran,

1958 → Algol,

1959 → COBOL,

1972 → C,

1996 → OCaml

Question 15

Partiellement correct

Note de 7,33 sur 10,00

Pour cet exercice, on se place dans le contexte du pseudo-code suivant :

```
var t : integer
    foo : array [0..1] of integer;

procedure shoot_my(x : Mode integer);
begin
    foo[0] := 3;
    t := 0;
    x := x - 2;
end;

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

Remplir les valeurs des variables **à la fin de l'exécution du programme** en fonction du mode de passage utilisé.

Avec un mode de passage par **valeur** :

- foo[0] = ✓ , foo[1] = ✓ , t = ✓

Avec un mode de passage par **valeur-résultat**, à la Algol W (la l-value dans laquelle est copiée la valeur résultat est évaluée *au retour de la fonction*) :

- foo[0] = ✗ , foo[1] = ✓ , t = ✓

Avec un mode de passage par **valeur-résultat**, à la Ada (la l-value dans laquelle est copiée la valeur résultat est évaluée *à l'appel de la fonction*) :

- foo[0] = ✓ , foo[1] = ✗ , t = ✓

Avec un mode de passage par **référence** :

- foo[0] = ✓ , foo[1] = ✗ , t = ✓

Avec un mode de passage par **nom** :

- foo[0] = ✗ , foo[1] = ✓ , t = ✓

[← Annonces](#)

Aller à...