

IFT1166 - Session Été 2000, Intra

Mohamed Lokbani

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date: 30 mai 2000

Durée: 2 heures (de 19h30 à 21h30) Local: P-310

Directives:

- Il vous est permis d'utiliser un livre de votre choix.
- Les documents de cours ne sont **pas** autorisés (**y compris mes e-notes disponibles sur la page web du cours ou à la bibliothèque Math/Info**).
- Ordinateurs personnels prohibés.
- Calculatrice (simple) permise.
- Répondre directement sur le questionnaire.

Bon courage!

Les niveaux de difficulté des questions posées:

* (la plus facile) ...à... *** (la plus difficile)

1. _____ /10

2. _____ /20

3. _____ /16

4. _____ /24

5. _____ /30

Total: _____ /100

Question 1 (10 points) (*)

Écrire la définition de la fonction `Qnum1`, qui n'accepte aucun argument et ne retourne aucune valeur, dont la tâche est d'afficher la figure suivante (représentant la lettre **i**). Il n'y a pas de contraintes sur la démarche à suivre, seul le résultat compte.

```
#####  
### ###  
#####  
### ###  
### ###  
### ###  
#####
```

Question 2 (20 points)

Q2.A Définissez ce que c'est le "polymorphisme", et donnez un exemple. (*)

Q2.B Définissez ce que c'est "une copie superficielle", et donnez un exemple. (*)

Q2.C Le destructeur a-t-il un type de retour? (*)

oui, il a un type de retour. ; **non**, il n'a pas un type de retour.

si la réponse est **oui**, lequel?

Q2.D Écrire le prototype de la fonction `exemple` qui retourne un pointeur sur un entier et qui accepte les 3 arguments suivants: une chaîne de caractères, un caractère passé par valeur, un entier passé par référence. (*)

Q2.E Soit la déclaration suivante: (**)

```
const char *chaine = "IFT1166";
```

Les instructions suivantes génèrent-elles une erreur à la compilation?

Q2.E.1. `*(chaine++)`;

oui, elle génère une err. de compil. ; **non**, elle ne génère pas une err. de compil.

justifiez votre choix (pourquoi)?

Q2.E.2. `(*chaine)++`;

oui, elle génère une err. de compil. ; **non**, elle ne génère pas une err. de compil.

justifiez votre choix (pourquoi)?

Question 3 (16 points) (**)

Tout en développant votre réponse sur la feuille d'examen, indiquer ce qu'écrit en sortie, le programme suivant:

```
// -----  
( 1) #include <iostream.h>  
  
( 2) int A = 2;  
( 3) int B = 3;  
  
( 4) void fonction()  
( 5) {  
( 6)     int B = 9;  
( 7)     while (B > 7)  
( 8)     {  
( 9)         int A = B;  
(10)         A++;  
(11)         B--;  
(12)     }  
  
(13)     cout << "A= " << A << " B= " << B << endl;  
  
(14) }  
  
(15) void uneAutre()  
(16) {  
(17)     int A = B;  
(18)     A++;  
(19)     B = 6;  
(20) }  
  
(21) int main()  
(22) {  
  
(23)     cout << "A= " << A << " B= " << B << endl;  
  
(24)     fonction();  
  
(25)     cout << "A= " << A << " B= " << B << endl;  
  
(26)     A = 11;  
(27)     int B = 12;  
  
(28)     uneAutre();  
  
(29)     cout << "A= " << A << " B= " << B << endl;  
  
(30)     return 0;  
(31) }  
// -----
```

A=	B=
----	----

A=	B=
----	----

A=	B=
----	----

A=	B=
----	----

Question 4 (24 points) (***)

Tout en développant votre réponse sur la feuille d'examen, indiquer ce qu'écrit en sortie, le programme suivant:

```
// -----  
( 1) #include <iostream.h>  
  
( 2) int uneFonction(int a, int & b,int *c)  
( 3) {  
( 4)   a += 1;  
( 5)   b += 3;  
( 6)   *c += 4;  
  
( 7)   return a+b+(*c);  
( 8) }  
  
( 9) int main() {  
  
(10)   int a = 3, b = 4,c=5;  
  
(11)   cout << "sortie:" << uneFonction(a,b,&c) << endl; 

|         |
|---------|
| sortie: |
|---------|

  
  
(12)   cout << "a= " << a << " b= " << b << " c= " << c << endl;  


|    |    |    |
|----|----|----|
| a= | b= | c= |
|----|----|----|

  
  
(13)   a = 3, b = 4, c=5;  
  
(14)   cout << "sortie:" << uneFonction(b,c,&a) << endl; 

|         |
|---------|
| sortie: |
|---------|

  
  
(15)   cout << "a= " << a << " b= " << b << " c= " << c << endl;  


|    |    |    |
|----|----|----|
| a= | b= | c= |
|----|----|----|

  
  
(16)   a = 3, b = 4,c=5;  
  
(17)   cout << "sortie:" << uneFonction(a,a,&a) << endl; 

|         |
|---------|
| sortie: |
|---------|

  
  
(18)   cout << "a= " << a << " b= " << b << " c= " << c << endl;  


|    |    |    |
|----|----|----|
| a= | b= | c= |
|----|----|----|

  
  
(19)   return 0;  
(20) }  
// -----
```

Question 5 (30 points)

Soit la fonction `main` suivante:

```
// -----  
( 1) int main() {  
  
( 2)     Question5 x; // chaîne vide  
  
( 3)     x.affiche();  
( 4)     x.affichageRenverse();  
  
( 5)     Question5 y("bateau");  
  
( 6)     y.affiche();  
( 7)     y.affichageRenverse();  
  
( 8)     Question5 z = y;  
  
( 9)     z.affiche();  
(10)     z.affichageRenverse();  
  
(11)     if (z.compare("bateau"))  
(12)         cout << "identique\n";  
(13)     else cout << "different\n";  
  
(14)     return 0;  
  
(15) }  
// -----
```

La fonction `main` utilise la classe `Question5`, qui contient entre autre les éléments suivants:

donnée membre private:

`char *nom` : le nom d'une chaîne de caractères.

fonctions membres public:

`affiche` : permet d'afficher sur la sortie standard, la donnée `nom`, membre de la classe `Question5`. Cette fonction n'accepte aucun argument, et elle ne retourne rien.

`affichageRenverse` : permet d'afficher la donnée `nom`, membre de la classe `Question5`, à l'envers (ex. pour la chaîne `soleil`, nous obtiendrons suite à l'appel de cette fonction, la chaîne `lielos`). Cette fonction n'accepte aucun argument, et elle ne retourne rien.

`compare` : une fonction booléenne permettant de comparer si la chaîne `nom` (membre de la classe `Question5`) est identique à la chaîne passée en argument. Cette fonction accepte donc, un seul argument du type chaîne de caractères et elle retourne `true`, si c'est vrai i.e. les deux chaînes sont identiques ; sinon `false` i.e. les deux chaînes sont différentes.

Q5.1 La classe `Question5` doit-elle avoir un Constructeur de Recopie (CR) explicite, ou celui par défaut est suffisant? (*)

oui, elle doit avoir un CR explicite ; **non**, elle ne doit pas avoir un CR explicite

pourquoi (justifiez votre réponse)?

Q5.2 Indiquer, parmi les lignes de la fonction `main` mentionnées dans le tableau ci-dessous, où est appelé le constructeur de recopie (explicite ou par défaut) de la classe `Question5`, en indiquant à chaque fois le nom de l'objet créé (si plusieurs objets sont créés pour une même ligne, indiquez-les tous). (*)

lignes (voir page 7) →	2	5	8	11
constructeur de recopie				

Q5.3 Écrire la définition de la classe `Question5`, telle que décrite précédemment. (***)

Q5.4 En utilisant la classe Question5 décrite au début de cet exercice, indiquez si la fonction main suivante compile correctement ou non. Si elle ne compile pas, indiquez pourquoi. (détaillez votre réponse)? (**)

```
// -----  
( 1) int main() {  
  
( 2)     Question5 y("bateau");  
( 3)     y.affiche();  
( 4)     y.affichageRenverse();  
  
( 5)     Question5 z = y;  
( 6)     z.affiche();  
( 7)     z.affichageRenverse();  
  
( 8)     if (z.compare(y.nom))  
( 9)         cout << "identique\n";  
(10)     else cout << "different\n";  
  
(11)     return 0;  
  
(12) }  
// -----
```

Q5.5 Une autre fonction `compare` est introduite dans la classe `Question5`. Définissez cette fonction `compare` (en respectant le principe de l'encapsulation), qui permet de comparer les noms de deux objets. Cette fonction accepte un seul argument du type `Question5` ; et elle retourne `true`, si les noms des deux objets sont identiques ; sinon `false`. La fonction main suivante montre un exemple d'utilisation de cette fonction `compare` (pas besoin de réécrire toute la définition de la classe `Question5`, uniquement les ajouts/modifications, s'il y en a).
(*)**

```
(1) int main() {  
(2)  
(3)     Question5 x("navire");  
(4)     Question5 y("bateau");  
  
(5)     if (x.compare(y))  
(6)         cout << "identique\n";  
(7)     else cout << "different\n";  
  
(8)     return 0;  
(9) }
```

