



Module : Programmation orienté objet (NI744)
M1 Instrumentation
Date : 30/01/2020

Responsable : I. Laribi
Année : 2020/2021
Durée : 1h30

Questions de cours (6p)

1) A quoi sert un compilateur ? (0.5)

R : un compilateur est un programme chargé de traduire ou transformer le code source d'un langage de programmation donné vers le langage binaire (programme exécutable).

2) Quelle est le rôle de la bibliothèque iostream dans un programme C++ ? (0.5)

R : Bibliothèque des entrée/Sortie cin et cout

3) Si on n'utilise pas l'expression using namespace std, que faut-il ajouté à ce code : (0.5)

R : `std :: cout << "Bonjour"<<end ;`

```
cout << "Bonjour"<<end ;
```

4) Pour tous les noms de variables suivants, choisissez ceux qui ne sont pas acceptés par le compilateur (avec justification) : (0.75)

include	var2	main	Var#2	2eme_element
---------	------	------	-------	--------------

R :

include , main → mots réservés.

Var#2 → # caractère non autorisé.

2eme_element → commence par un chiffre.

5) Quels sont les deux types de commentaires admis en C++ ? En quoi diffèrent-ils ? (0.5)

R : // ceci est un commentaire → commentaire sur une seul ligne

/* ceci est un → commentaire sur une ou plusieurs lignes
 Commentaire */

6) Comment choisir entre l'instruction if...else et l'instruction switch ? (0.5)

R : switch pour éviter if ... else ... imbriquées.

7) Quelle est la différence entre if (x=3) et if (x==3) ? (0.5)

R : if (x=3) affecte la valeur 3 et renvoie la valeur 3, qui est considéré comme vrai.

if (x==3) teste si x est égale a 3 ; elle renvoie vrai si la valeur de x est égale à 3 et faux dans le cas contraire.

8) Quelle différence y a-t-il entre un prototype de fonction et une définition de fonction ? (0.5)

R : Le prototype de la fonction déclare la fonction, la définition la définit. Le prototype se termine par un point-virgule, pas la définition.

Le prototype décrit l'entête de la fonction avant son premier appel.

Définition de la fonction =entête + corps de la fonction.

9) Qu'est-ce qu'une classe dans la POO ? (0.5)

R : une classe décrit la structure interne d'un objet : les données qu'il regroupe, les actions qu'il est capable d'assurer sur ces données.

10) Quelle est la bonne expression qui permet de créer un objet de type Point ? (0.5)

- A. Point=a ;
- B. Point a ;
- C. Point int a ;

11) Chercher L'erreur : pourquoi ce code est-il bogué ? (0.75)

```
int compteur= 0;
while (compteur <10)
{
cout <<"compteur : " <<compteur ;
}
```

R : le compteur n'est jamais incrémenté, la boucle While ne se termine jamais.

Exercice 1 : (7.75p)

Soit la classe **Compte** qui contient une donnée membre **solde** de type entier et trois méthodes membres :

Une méthode **déposer** qui ne retourne rien et possède une entrée **montant** de type entier, cette méthode permet d'ajouter le contenu du montant au solde.

Une méthode **retirer** qui ne retourne rien et possède une entrée **montant** de type entier, cette méthode permet de soustraire du solde la valeur du montant.

Une méthode **afficher** qui ne retourne rien et n'a aucun paramètre en entrée, cette méthode permet d'afficher la valeur du solde sur écran.

1. Ecrire le programme C++ qui définit la classe Compte.
2. Dans la partie main (), créer deux Compte C1, C2 et écrivez le code correspondant aux opérations suivantes :
 - Dépôt de 5000 dinars sur le compte C1.
 - Dépôt de 10000 dinars sur le compte C2.
 - Retrait de 3000 dinars sur le compte C2.
 - Afficher les soldes des comptes C1 et C2.

N.B.

- Le solde des deux comptes au départ est de 0 dinars.

- Les méthodes sont à définir à l'extérieur de la classe.

- Toutes les membres de classe sont Public.

```
#include <iostream>
using namespace std;
```

class Compte {	0.5
public:	0.25
int solde;	0.25
public:	
void deposer(int montant);	0.25
void retirer (int montant);	0.25
void afficher();	0.25
};	0.25
void Compte::deposer(int montant) { solde=solde+montant; }	1
void Compte::retirer(int montant) { solde=solde-montant; }	1
void Compte::afficher() { cout <<solde<<endl; }	1
int main() {	0.25
Compte C1,C2;	0.5
C1.solde=0; C2.solde=0;	0.25
C1.deposer(5000);	0.5
C2.deposer(10000);	
C2.retirer(3000);	0.5
C1.afficher();	
C2.afficher();	0.25
}	

Exercice 2 : (6.25p)

Ecrire un programme qui demande à l'utilisateur de taper des entiers strictement positifs et qui affiche le nombre de valeurs multiples de 5 tapées.

Lorsqu'on tape une valeur négative, le programme affiche ERREUR et demande de retaper une valeur.

Lorsqu'on tape 0, cela signifie que le dernier entier a été tapé (c'est-à-dire, que le programme s'arrête lorsqu'on lui fournit la valeur 0). On affiche alors le nombre de valeurs multiples de 5, par contre, si le nombre d'entiers tapés est égal à 0, on affiche : PAS DE VALEURS A AFFICHER.

#include <iostream>	
using namespace std;	0.25
int main() {	
int n,compteur=0;	0.75
cout <<"tapez un entier positive"<<endl;	
cin>>n;	0.5
while (n!=0) {	1
if (n<0)	0.5
cout<<"ERREUR, retapez un autre entier"<<endl;	0.25
else if (n%5==0)	1
compteur =compteur+1;	0.5
cin>>n; }	0.25
if (compteur ==0) cout <<"pas de valeur a afficher"<<endl;	0.75
else cout <<"le nombre de valeurs multiples de 5 est : "<<compteur<<endl; }	0.5