

Durant cette séance, vous allez :

- Comprendre la notion du Constructeur/Destructeur et son intérêt
- Manipuler l'allocation dynamique et ses avantages
- Connaître comment allouer et libérer la mémoire

1 Appel du constructeur /Destructeur

Définir la classe fête afin d'aboutir à cette exécution :

```
int main()
{
    Fete f;
    cout << "super!" << endl;
    f.inviter();
    cout << "Non je dois rentrer che moi!" << endl;
    return 0;
}
```

```
"E:\Esprit\Mini Projet C\code_block_WorkSpace\fete\bin\Debug\fete.exe"
La fete a commenc 
super!
Tu va rester?
Non je dois rentrer che moi!
Ok! Au revoir
```

2 Ordre d'appel

Donner la sortie du programme suivant dans chaque cas et commenter bri vement:

```
#include <iostream>

class Truc{

public:

    Truc() { cout << "+++++" << endl; }

    ~Truc() { cout << "-----" << endl; }

};
```

1. void main() {Truc x;}
2. void main() {Truc *x=new Truc();}
3. void main() {Truc * x=new Truc(); delete x;}

3 Allocation dynamique d'un tableau

Donner la sortie de chaque étape et commenter brièvement:

```
using namespace std;
class Voiture
{
private:
    string couleur, marque;
public:
    Voiture() {cout <<"je suis le constructeur de la classe Voiture"<<endl;};
    ~Voiture() {cout <<"je suis le destructeur de la classe Voiture"<<endl;};
};

int main()
{
    //étape 1
    Voiture *tab =new Voiture[2];
    cout <<"*****"<<endl;
    /*étape 2
    delete [] tab;
    */
    return 0;
}
```

4 Classe Note

Créer un nouveau fichier entête notes.h et un fichier source notes.cpp contenant la classe suivante :

Notes
- test 1 : float - test 2 : float - DS: float - Examen : float - orale : float - TP : float - avecTP :bool
+ Notes() + Notes(float,float,float,float,float,float) + Notes(float,float,float,float,float) - calculNCC() : float + calculMoyenne() : float

Créer cette classe sachant que :

- (a) Tous les attributs sont privés. Toutes les opérations sont publiques sauf *calculNCC*.
 - (b) Déclarer un constructeur par défaut plaçant 0 dans toutes les notes et vrai dans *AvecTP*.
 - (c) Ajouter un deuxième constructeur avec des paramètres permettant d'initialiser les différentes notes y compris la note de TP.
 - (d) Créer un troisième constructeur avec des paramètres permettant d'initialiser les notes sans note de TP. Ce constructeur va modifier la valeur de l'attribue avecTP à faux.
 - (e) Implémenter les opérations CalculNCC et CalculMoyenne sachant qu'elle s'effectue de la façon suivante :
 - i. $NCC = (teste\ 1 + teste2 + orale + 2 * DS) / 5$
 - ii. S'il y a TP : $Moyenne = 30\% NCC + 20\% TP + 50\% Examen$
 - iii. Sinon : $Moyenne = 40\% NCC + 60\% Examen$
2. Créer le programme principal permettant de déclarer N 1 et N2 et d'afficher la moyenne la plus grande. Avec :

```

||
N 1: 10 , 12 , 15 , 11 , 17 , 9.5
N 2: 11.5 , 13 , 18 , 10 , 12.5
  
```

5 Implementation des classes

On se propose de développer une application destinée à une agence des voyages organisés afin de gérer les réservations effectuées par ses clients.

Chaque voyage organisé est caractérisé par un identifiant unique, une ville d'origine, une ville de destination, une date de départ, une date de retour, un prix, le nom et un hébergement.

Un hébergement est caractérisé par l'adresse, nom et nombre des étoiles de l'hôtel.

✓ Créer deux hébergements

« Boulevard de la Promenade, 8050 Hammamet, Tunisie », 5, « Yasmine Beach »

« BP437 / Hammamet 8050 / Tunisie », 2, « la Badira Hammamet »

✓ Créer un voyage.

- ✓ Affecter un hébergement à ce voyage.
- ✓ Accéder à ce voyage et modifier le nombre des étoiles à 1.
- ✓ Afficher les détails de ce voyage avec les détails de son hébergement (sans passer par la méthode afficher)