

ISIMA 2^{ème} année - Examen de C++ - durée 2 heures

1. **Généricité : (2 pts)** Ecrire les fonctions génériques qui permettent de trouver le maximum et le minimum entre deux instances d'un même type. Si les types sont des classes d'objets quelle méthode devront fournir ces classes ?

2. **Polymorphisme : (3 pts)**

Soit le code C++ suivant, donner le résultats des affichages :

```
#include <iostream>

class Shape {
    . . .
    void draw(){ cout << 1 << endl; }
};

class Circle : public Shape {
    . . .
    void draw() { cout << 2 << endl; }
};

class Square : public Shape {
    . . .
    void draw() { cout << 3 << endl; }
};

int main(void)
{
    Shape *s = new Shape();
    Circle *c = new Circle();
    Shape *s2 = c;

    s->draw(); // Affichage ???
    c->draw(); // Affichage ???
    s2->draw(); // Affichage ???

    return 0;
}
```

Que suggérez vous d'ajouter à la déclaration de la classe Shape ? Pourquoi ?

Comment s'exécute alors le code, est-ce que cela change les affichages ?

3. **Problème (10 pts)**

On désire concevoir une classe Histogramme destinée à la manipulation... d'histogrammes. On suppose que les données en entrée sont comprises entre les valeurs A et B. On souhaite disposer d'un constructeur acceptant en paramètres A, B et N, le nombre d'intervalles à prendre en compte.

- Quels sont les attributs nécessaires à cette classe ?
- Coder cette classe (déclaration et définition) avec ses attributs et l'ensemble minimal de méthodes qu'elle doit posséder.
- Définir une méthode d'affichage des valeurs d'un histogramme. Fournir ensuite la même fonctionnalité avec la sortie sur un ostream.
- Définir une méthode de mise à jour de l'histogramme après l'ajout d'une donnée.
- On suppose qu'on dispose d'une classe Echantillon qui est chargée de contenir un ensemble de données. On souhaite remplir l'histogramme à partir d'un échantillon,

sachant que les classes Echantillon et Histogramme ne sont pas supposées se connaître a priori. Proposer une interface pour la classe Echantillon et fournir le code pour la construction et l'affichage un histogramme à partir d'un échantillon.

4. Héritage multiple (2 pts)

Soit le code C++ suivant :

```
#include <iostream>

class Vehicule {
public:
    int moteur;
};

class Avion : public Vehicule {
public:
    Avion() { moteur=0; }

    void voler();
};

class Bateau : public Vehicule {
public:
    Bateau() { moteur=1; }

    void flotter();
};

class Hydravion : public Avion, public Bateau {
};
```

Ecrire une fonction principale main qui affiche sur "cout" la valeur des moteurs d'un hydravion que vous aurez instancié dans cette fonction main.

Comment modifier ce code pour éviter l'héritage à répétition ?

5. STL (3 pts)

Commenter de manière précise le code suivant :

```
#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <vector>
#include <iterator>
#include <algorithm>

using namespace std;

int main(void) {
    typedef vector<int> vector_int;

    ifstream source("input.txt");

    vector_int v;
    istream_iterator<int> start(source);
    istream_iterator<int> end;
    back_inserter_iterator<vector_int> dest(v);

    copy(start, end, dest);
    sort(v.begin(), v.end());
    copy(v.begin(), v.end(), ostream_iterator<int>(cout, "-"));

    cout << endl << "Fin du test !" << endl;

    return 0;
}
```