Université Djilali Liabes
Faculté du Génie Électrique
Département d'électronique
Master II Systèmes embarqués
Module : Programmation JAVA
Durée : 1 h 30

Partie I (Compilation 05 Pts)

- Quelle est la définition d'un interpréteur (01 Pt)
  Un interpréteur est un utilitaire généralement fournit avec les systèmes d'exploitation à l'instar du Shell qui permet d'interpréter des instructions sans générer de programme exécutable
- Quel est le rôle des directives de préprocesseurs (01 Pt)
  Le rôle des directives de préprocesseur est d'indiquer par programme les directives que le compilateur se doit de suivre (Compilation sur un système d'exploitation particulier , du matériel spécifique , des remplacements à effectuer avant d'initier la phase de compilation, ...ETC)
- Expliquez à l'appui d'un synopsis les phases de compilation (03 pts)
  L'analyse lexicale : Vérifier que le texte fourni en entrée peut se découper en mots (unités lexicales qui seront utilisées dans la phase suivante), de sorte que chaque mot appartienne au langage utilisé.
  L'analyse syntaxique : Vérifier que le flux de tokens généré par l'analyseur lexical forme une succession de phrases conforme à la syntaxe du langage utilisé
  L'analyse sémantique : L'objectif de cette phase est de vérifier que toutes les phrases d'ecrites dans l'analyse syntaxique ont un sens (elles veulent dire quelque chose) dans le langage utilisé.
  L'édition de liens : utiliser des librairies de fonctions standard déjà écrites comme les fonctions d'affichage par exemple). Une fois le code source assemblé, il faut donc lier entre eux les différents fichiers objets.
  L'optimisation : Réduction de la complexité des programmes objets et optimisation autant que possible des instructions dans l'optique d'optimiser les ressources et les cycles processeurs necessères pour l'exécution du programme
  La génération du code : Le générateur de code utilise généralement les analyses fourni par l'analyseur syntaxique pour effectuer cette traduction et générer un code objet.

Partie II (Généralités JAVA 05 Pts)

- Quelles sont les étapes pour l'écriture d'un programme JAVA (01 Pt)
  Écriture : production d'un code dans le langage JAVA
  Compiler le programme
  - *Traduire le programme dans un langage de bas niveau (machine)*
  - *[éventuellement optimisation]*
  - *Produire un programme (code) exécutable*

  Exécution
  - *Charger le programme en mémoire (typiquement en tapant le nom du programme exécutable)*
  - *Exécution*

Université Djilali Liabes
Faculté du Génie Électrique
Département d'électronique
Master II Systèmes embarqués
Module : Programmation JAVA
Durée : 1 h 30

- Quelle est la différence entre la syntaxe et la sémantique d'une instruction (01 Pt)
  La syntaxe est reconnue par le langage alors que la sémantique est davantage liée au sens de l'instruction
- Quelle extensions sont utilisées pour les fichiers sources et les fichiers compilés en JAVA (01 Pt)
  Respectivement .java .class
- Que représente respectivement une classe et un objet JAVA (01 Pt)
  La classe et la structure qui permet la production d'un objet, en d'autre termes l'objet est une instance de class
- Quelle est la structure de base d'un objet (01 Pt)
  Un objet étant une instance de classe , sa structure de base est donc composée d'un constructeur , destructeur , des attributs et des méthodes

Partie III (Programmation de base JAVA 05 Pts)

- Concevez et proposer une implémentation d'une classe JAVA qui calcule l'accumulateur (la somme des éléments) d'un vecteur de double de taille 10 (02 Pts)

```
class Acuum
{
  static void main(String[ ] args)
  {
   double  [ ] table= {10,11,12,13,-6,6,4,78,2,-5.2};
   double accum =0;
   for ( i = 0 ; i<10 ; i++ )
     accum=accum+table[i];
  }
```

- Concevez et proposer une implémentation d'une classe JAVA qui réalise l'adition de deux vecteur (élément par élément) de deux vecteur de double de taille 10 (03 Pts)

```
class Acuum
{
  static void main(String[ ] args)
  {
   double [ ] VectA= {10,11,12,13,-6,6,4,78,2,-5.2};
   double [ ] VectB= {25,1,21,-3.5,-66,66,44,56,20,-250.3};
  double [ ] VectC= {0,0,0,0,0,0,0,0,0,0};
   for ( i = 0 ; i<10 ; i++ )
     VectC[i]=VectA[i]+VectB[i];
  }
```

Partie IV (Programmation Orienté objet en JAVA 05 Pts)

A)

Soit l'implémentation des deux classes suivantes :

```
class A {
        public void meth() { System.out.println("Salut"); }
}
class B extends A {
```

Université Djilali Liabes
Faculté du Génie Électrique
Département d'électronique
Master II Systèmes embarqués
Module : Programmation JAVA
Durée : 1 h 30

```
        int var;
    }
```

Un programmeur pour manipuler les deux classes a utilisé le programme suivant qui refuse de compiler

```
A a = new A();
B b = new B();
a = b;
a.var = 2;
```

Question 1 : Pour quelle raison le code du programmeur refuse t-il de compiler (01 Pt)

A et B ne sont pas du même type

Question 2 : Proposer une technique n qui permettra au programme de compiler (01 Pt)

A traves une opération de casting (transtypage) il est possible de résoudre ce problème

Question 3 : Écrire un programme en java qui permet d'ouvrir, de traiter et d'enregistrer un fichier (03 Pts)

```
public static void ouvrir_fichier(String nom) {
    try {
      input = new BufferedReader(
                  new FileReader(nom));
    }
    catch (IOException e) {
      System.err.println("Impossible  d'ouvrir le fichier d'entree.\n" + e.toString());
      System.exit(1);
    }
}
public static void traiter_fichier() {
    String ligne;
    try { // catch EOFException
      ligne = input.readLine();
      while (ligne != null) {
          System.out.println(ligne);
          ligne = input.readLine();
      }
}
 public static void fermer_fichier() {
    try {
      input.close();
      System.exit(0);
    }
    catch (IOException e) {
      System.err.println("Impossible de fermer les fichiers.\n" + e.toString());
    }
}
```