Assertion :

L'objectif de ce TD est d'apprendre et comprendre la notion de la classe (types, attributs, actions, et l'instanciation d'un objet), ainsi que les instructions d'affichage, et l'exécution d'un code de JAVA simple.

Exercice 1:

1. Quelle est la sortie du fragment de code suivant ? C'est quoi la méthode **« main» ?**
2. Sert-il à quoi le paramètre **« args »** ?

```
1. public class ClassExample
2. {
3. public static void main(String[] args)
4. {
5. Int i= 5;
6. System.Out.Println(i+"");
7. }
8. }
```

R1.

« main » est la méthode principale à partir de laquelle le code Java s'exécute. Cette méthodedoit obligatoirement existe dans n'importe quel projet Java, et elle a une seule définition,ouonpasse des arguments sous forme d'un tableau de chaine de caractères.

R2.

« args » est un ensemble des paramètres qu'on passe à la méthode main, et sont de type chaine de caractères. Ces paramètres sont spécifiables sur les IDEs Java (via une fenêtre de configuration.), ou même par une exécution via la ligne de commande.

Exercice 2:

1. Quelle est la différence entre un classe **«abstract », «static»**, et **«concret»** ? Donner des exemples simples.
2. Que signifie le mot clé «**protected**», et «**private**» devant un variable en Java ?

R1.1

Une classe abstraite est une classe qu'on ne peut pas créer des objets ou définir un constricteur, elle contient que des variables, et des méthodes abstraites.

R1.2

Le mot clé « static » est utilisé pour définir une structure de donnée (variable, méthodes, ou classe) qui caractérise la classe globale. On dit qu'une structure de donnée statique est partageable entre tous les objets de la classe globale.

- **Variable statique :** un variable statique définit la classe, et pas les objets de la classe. On prenant un exemple concret comme la classe Etudiant, le variable «**nom_class**» caractérise la classe et il a une valeur constante «Etudiant», d'une

manière similaire on peut aussi définir le variable «**nombre_instances**» qui représente le nombre des objets de la classe.

- **Méthode statique:** les méthodes statiques caractérisent la classe, et sont applicables que sur des variables statiques.
- **Classe statique:** les classes statiques sont toujours internes, et elles ont une manière d'instanciation différente, ou on doit obligatoirement spécifier chaque fois on crée un objet, la classe globale (laquelle contient la classe statique). Voyant l'exemple :

```
Class Global{
 // some variables
// some constructors
// some methods
static class NestedStaticClass{

// some variables

// some constructors
// some methods
}
}
Class Main{
public static void main(String[] args){
Global.NestedStaticClass nc = new Global.NestedStaticClass();
}
}
```

R.2

Un variable privé est accessible par les méthodes de la classe elle-même seulement, par contre un variable protégé est accessible à partir de la classe et ses dérivés en héritage.

Exercice 3 :

Sachant que les tableaux en Java se définit par des crochets '[]' comme l'exemple :

```
1. int[] notes ;
```

1. Définir la classe **étudiant**, en spécifiant le nom, le prénom, et les notes. Ajouter les méthodes permettant de mettre, et de lire le nom, le prénom, et les notes, ainsi que la méthode permettant de calculer la moyenne de note.
2. Créer une instance de cette classe en spécifiant (nom : votre nom, prénom : votre prénom, notes : 10 valeurs entiers aléatoires).

R4:

Etudiant.java :

```
class Etudiant{
// some variables
String nom,prenom ;
float[] notes;
```

```
some constructor
public Etudiant(String nom, String prenom, float[] notes){
this.nom= nom ;
this.prenom=prenom ;
this.notes=notes;
}
public float computeAverage(){
//some calculation
        return avg ;
}
public String toString(){
// some formatting
 return me;
}
```

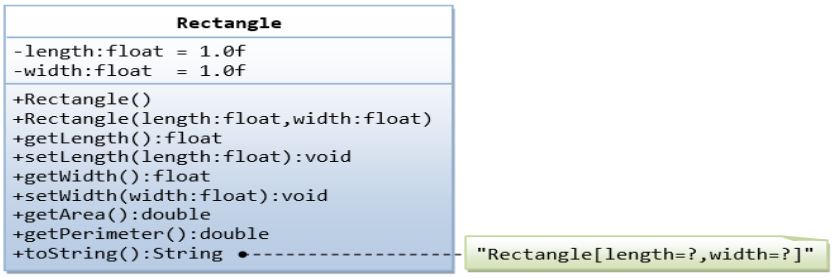
Main.Java

```
class Main {
public static void main(String[] args){
        Etudiant et1 = new Etudiant("";"",notes);
        float avg = et1. ComputeAverage() ;
        System.out.println(et1);
}
}
```

Code Java sera partagé sur la Platform codé par un de vos camarades.

Exercise 4:

1. Soit la figure suivante, écrivez le code java équivalent.



2. C'est quoi un constricteur d'une classe ?

R4.1

Rectangle.java

```
class Rectangle{
flaot width, length ;
public Rectangle(){
```

```
}
public Rectangle(flaot width, float length){
this.width= width ;
this.length=length ;
}
public getLength(){
}
public setLength(float length){
}
public getWidth(){
}
public setWidth(float length){
 }
public float getArea(){
}
public float getPerimeter(){
}
public String toString(){
}
}
```

Main.java

```
public class main{
public static void main(String[] args){
        ///
}
}
```

Code Java sera partagé sur la Platform codé par un de vos camarades.

R4.2

Le constricteur est une méthode spéciale qui fait la construction d'un objet de classe (l'allocation de l'espace mémoire de chaque variable de la classe, l'initialisation …etc.**).**

Exercice 5:

Sachant qu'une matrice en Java se défini comme l'exemple suivant :

```
1. int[][] matrix ;
```

On veut créer un code java permettant de calculer la somme de deux matrices, ou chaque matrice est définie par ses deux dimensions m, n, et une étiquette.

1. Ecrivez le code java qui prend en entré deux matrices « M » et « N », et renvoie leurs somme. Assurer que votre code respecte les principes du POO.

R5.1

En terme de POO, une classe est définie par deux dimension N, et M, et un tableau contenant les valeurs de la matrice. En plus, une classe matrice peut avoir un constricteur d'initialisation, et des actions de calcule mathématique / algébrique (addition,

soustraction, multiplication, déterminent, l'inverse ...). Un code d'un de vos collègues sera partagé sur la Platform.

Exercice 6 : -à faire à la maison-

Soit le jeu connu suivant :

X	O	O
X	O	
X		

1. Proposer une solution POO qui permet de dérouler ce jeu (classes, et actions …). On s'intéresse par le coté logique seulement.

Un projet de vos camarades sera partagé par la fin de ce semestre.