

TP 3: CLASSES ET OBJETS

Exercice1

```
public class Rectangle {  
    float longueur, largeur;  
    public Rectangle()  
    {  
    }  
    public Rectangle(float longueur, float largeur) {  
        this.longueur = longueur;  
        this.largeur = largeur;  
    }  
    public float perimetre() {  
        return 2 * (longueur + largeur);  
    }  
    public float surface() {  
        return longueur * largeur;  
    }  
}
```

Dans la fonction statique *main()* on ajoute :

```
Rectangle rec1= new Rectangle(12,3.4);
```

Exercice2

```
public class Cercle {  
    float r;  
    // constructeurs  
    public Cercle(){// par défaut  
  
    }  
    public Cercle(float r){//avec  
        paramétres this.r=r;  
    }  
    public float Perimetre(float r){  
        return 2*(float) Math.PI*r;  
    }  
    public float Surface(float r){  
        return (float) Math.PI*(r*r);  
    }  
    public static void main(String[]  
args) { Cercle C1=new Cercle(); C1.r=5;  
  
Cercle C2=new Cercle(3);  
float S1= C1.Surface(5);  
float S2= C2.Surface(3);  
float Minimum= Math.min(S1, S2);  
float P1= C1.Perimetre(5);  
float P2= C2.Perimetre(3);  
float Maximum= Math.max(P1, P2);  
System.out.println("minimum="+Minimum+"Maximum="+Maximum); }  
}
```

a. **Oui**, on peut utiliser les méthodes périmètre et surface sans passer de paramètres.

Car on crée l'objet soit par le constructeur 02 donc on initialise le **rayon r** , soit par le constructeur par défaut et on attribue une valeur de **r** . Donc, inutile de passer **r** en argument.

Exercice 03:

```
public class Employe {
    static int compteur=0;
    String nom,pernom,fonction;
    float salaire;
    public Employe(){
        compteur++;
    }
    public Employe(String N,String P ){
        this.nom=N;
        this.pernom=P;
    }
    public Employe(String N,String P,String F ){
        compteur++;
        this.nom=N;
        this.pernom=P;
        this.fonction=F;
    }
    public Employe(String N,String P,String F, float S){
        compteur++;
        this.nom=N;
        this.pernom=P;
        this.fonction=F;
        this.salaire=S;
    }

    public String toString() {
        return "Employe{" + "nom=" + nom + ", pernom=" + pernom + ", fonction=" +
fonction + ", salaire=" + salaire + '}';
    }

    public static void main(String [] args){
        Employe P1= new Employe();
        P1.nom="ahmed";
        P1.pernom="said";
        P1.fonction="administratif";
        P1.salaire=30000;
        String sp=P1.toString();
        System.out.println(sp);
    }
}
```

Classe TraitementEmploye.class:

```
package personne;

public class traitementEmploye {
    public static void main (String[]args){
        Employe P1=new Employe();
        P1.nom="ahmed";
        P1.pernom="said";
        P1.fonction="administratif";
        P1.salaire=30000;
        String sp=P1.toString();System.out.println(sp);//ou bien
        directement System.out.println(P1.toString())
    }
}
```

- La condition est: les deux classes **Employe** et **TraitementEmploye** sont dans **le même package**