

**Module : Programmation C++**  
**Filière Mécatronique / 1<sup>er</sup> Année / Année 2017-2018**

**Corrigés TD/TP N° 3**

**Exercice 1**

Trouver les erreurs dans les déclarations des classes suivantes :

```
(A) class Circle:
{
    private
        double centerX;
        double centerY;
        double radius;
    public
        setCenter(double, double);
        setRadius(double);
}
```

**// Réponse**

```
class Circle // supprimer :
{
    private :
        double centerX;
        double centerY;
        double radius;
    public :
        void setCenter(double, double);
        void setRadius(double);
};
```

```
(B) #include <iostream>
using namespace std;
Class Moon;
{
    Private;
    double earthWeight;
    double moonWeight;
    Public;
    moonWeight(double ew);
    { earthWeight = ew; moonWeight = earthWeight / 6; }
    double getMoonWeight();
    { return moonWeight; }
}
int main()
{
```

```

double earth;
cout >> "What is your weight? ";
cin << earth;
Moon lunar(earth);
cout << "On the moon you would weigh "
      << lunar.getMoonWeight() << endl;
return 0;
}

```

### **// Réponse**

```

#include <iostream>
using namespace std;
class Moon // supprimer ;
{
private:
    double earthWeight;
    double moonWeight;
public:
    Moon(double ew) // supprimer ;
        { earthWeight = ew; moonWeight = earthWeight / 6.0; }
    double getMoonWeight() // supprimer ;
        { return moonWeight; }
};
int main()
{
    double earth;
    cout << "What is your weight? ";
    cin >> earth;
    Moon lunar(earth);
    cout << "On the moon you would weigh "
          << lunar.getMoonWeight() << endl;
    return 0;
}

```

```

(C) #include <iostream>
using namespace std;
class DumbBell;
{
    int weight;
public:
    void setWeight(int);
};
void setWeight(int w)
{
    weight = w;
}
int main()
{
    DumbBell bar;
    DumbBell(200);
    cout << "The weight is " << bar.weight << endl;
    return 0;
}

```

## Module : Programmation C++ Filière Mécatronique / 1<sup>er</sup> Année / Année 2017-2018

**// Réponse**

```
#include <iostream>
using namespace std;
class DumbBell // supprimer ;
{
    int weight;
public:
    void setWeight(int);
};
void DumbBell::setWeight(int w)
{
    weight = w;
}
int main()
{
    DumbBell bar;
    DumbBell(200); // à supprimer
    /* car il n'y a pas de constructeur avec argument
       Pour cette classe */
    cout << "The weight is " << bar.weight << endl;
    /* à supprimer, car weight est une donnée membre
       privée */
    return 0;
}
```

```
(D) class Change
{
public:
    int pennies;
    int nickels;
    int dimes;
    int quarters;
    Change()
        { pennies = nickels = dimes = quarters = 0; }
    Change(int p = 100, int n = 50, d = 50, q = 25);
};
void Change::Change(int p, int n, d, q)
{
    pennies = p;
    nickels = n;
    dimes = d;
    quarters = q;
}
```

**// Réponse**

```

class Change
{
    public:
        int pennies;
        int nickels;
        int dimes;
        int quarters;
        Change()
            { pennies = nickels = dimes = quarters = 0; }
        Change(int p = 100, int n = 50, int d = 50, int q = 25);
};
Change::Change(int p, int n, int d, int q)
/* il faut supprimer void, car il s'agit d'un
constructeur */
{
    pennies = p;
    nickels = n;
    dimes = d;
    quarters = q;
}

```

## **Exercice 2**

1. Ecrire une classe « Cercle » ayant une donnée membre privée « rayon » (double) et les 7 fonctions membres publiques suivantes :
  - Un constructeur par défaut qui initialise le rayon du cercle à 0.0.
  - Un constructeur qui accepte le rayon du cercle comme argument.
  - « setRayon » qui permet de modifier le rayon du cercle.
  - « getRayon » qui retourne le rayon du cercle.
  - « getSurface » qui retourne la surface du cercle en utilisant la formule  $surface = 3.14159 * rayon * rayon$ .
  - « getDiametre » qui retourne le diamètre du cercle, lequel est calculé par la formule  $diamètre = 2 * rayon$ .
  - « getPerimetre » qui retourne le périmètre du cercle, lequel est calculé par la formule  $Périmètre = 2 * 3.14159 * rayon$ .
2. Ecrire un programme principal pour tester la classe cercle en demandant tout d'abord à l'utilisateur de saisir le rayon du cercle, puis en créant un objet cercle, et ensuite en affichant la surface, le diamètre et le périmètre de ce cercle.
3. Modifier cette classe par l'ajout d'une donnée membre « pi » qui est un double initialisée à la valeur 3.14159, et d'un constructeur de copie.

***// Réponse de la question 1 (la classe) + question2 (programme de test)***

```

#include <iostream>
using namespace std;
class cercle
{

```

## **Module : Programmation C++**

### **Filière Mécatronique / 1<sup>er</sup> Année / Année 2017-2018**

**private:**

**double rayon;**

**public:**

**cercle();**

**cercle(double r);**

**void setRayon(double r);**

**double getRayon() const;**

**double getSurface() const;**

**double getDiametre() const;**

**double getPerimetre() const;**

**};**

**cercle::cercle()**

**{**

**rayon = 0.0;**

**}**

**cercle::cercle(double r)**

**{**

**rayon = r;**

**}**

**void cercle::setRayon(double r)**

**{**

**rayon = r;**

**}**

**double cercle::getRayon() const**

**{**

**return rayon;**

**}**

**double cercle::getSurface() const**

**{**

**return 3.14159 \* rayon \* rayon;**

**}**

```

double cercle::getDiametre() const
{
    return 2 * rayon;
}
double cercle::getPerimetre() const
{
    return 2 * 3.14159 * rayon;
}
int main()
{
    double r;
    cout << "Donner la valeur du rayon : ";
    cin >> r;
    cercle C(r);
    cout << "La surface du cercle est : " << C.getSurface() << endl;
    cout << "Le diametre du cercle est : " << C.getDiametre() << endl;
    cout << "Le perimetre du cercle est : " << C.getPerimetre() << endl;
    system("pause");
    return 0;
}

```

*// Réponse de la question 3 (Modification de la classe) + programme de test*

```

#include <iostream>
using namespace std;
class cercle
{
private:
    double rayon;
    static double pi; // donnée membre statique
public:
    cercle();
    cercle(double r);
    cercle(const cercle &); // constructeur de copie
    void setRayon(double r);
    double getRayon() const;
    double getSurface() const;
    double getDiametre() const;
}

```

## Module : Programmation C++ Filière Mécatronique / 1<sup>er</sup> Année / Année 2017-2018

```
double getPerimetre() const;  
};  
double cercle::pi = 3.14159;  
// L'initialisation de la donnée membre statique se fait hors la classe  
cercle::cercle()  
{  
    rayon = 0.0;  
}  
cercle::cercle(double r)  
{  
    rayon = r;  
}  
cercle::cercle(const cercle &c)  
{ // la définition du constructeur de copie  
    rayon = c.rayon;  
}  
void cercle::setRayon(double r)  
{  
    rayon = r;  
}  
double cercle::getRayon() const  
{  
    return rayon;  
}  
double cercle::getSurface() const  
{  
    return pi * rayon * rayon;  
}  
double cercle::getDiametre() const  
{  
    return 2 * rayon;
```

```
}  
double cercle::getPerimetre() const  
{  
    return 2 * pi * rayon;  
}  
int main()  
{  
    double r;  
    cout << "Donner la valeur du rayon : ";  
    cin >> r;  
    cercle C(r);  
    cout << "La surface du cercle C est : " << C.getSurface() << endl;  
    cout << "Le diametre du cercle C est : " << C.getDiametre() << endl;  
    cout << "Le perimetre du cercle C est : " << C.getPerimetre() << endl;  
    cercle D(C); // appel du constructeur de copie  
    cout << "La surface du cercle D est : " << D.getSurface() << endl;  
    cout << "Le diametre du cercle D est : " << D.getDiametre() << endl;  
    cout << "Le perimetre du cercle D est : " << D.getPerimetre() << endl;  
    system("pause");  
    return 0;  
}
```