

Object-Oriented Programming in C++

Exercise 01

Write in C++ a class "Rectangle" with two variables "a" and "b" and a member function "surface()" which will return the area of the rectangle.

Solution :

```
#include <iostream>
using namespace std;

class Rectangle {
    public:
    int a,b;
    int surface(){
        return a*b;
    }
};

int main() {
    Rectangle rectangle;
    cout<<"Entrez la largeur(a) du rectangle : "<<endl;
    cin>>rectangle.a;
    cout<<"Entrez la longueur(b) du rectangle: "<<endl;
    cin>>rectangle.b;

    cout<<"Surface : "<< rectangle.surface();
    system("PAUSE");
    return 0;
}
```

Exercise 02 :

Écrire une classe C++ appelée « Etudiant » avec les membres suivant :

nom : (de type char),

note1, note2 : (de type float)

calc_moy() : calcule la note moyenne.

afficher () : affiche le nom et la note moyenne.

Le programme principal (main) demande à l'utilisateur d'entrer le nom et les notes d'un étudiant. et affiche leur nom et la note moyenne.

Solution :

```
#include <iostream>
#include <string>
using namespace std;

class Etudiant {
public:
    string nom;
    float note1, note2;

    // Utilisez la liste d'initialisation du constructeur pour initialiser les membres
    // de la classe.
    Etudiant(string _nom, float _note1, float _note2) : nom(_nom), note1(_note1),
    note2(_note2) {}

    float calc_moy() const {
        return (note1 + note2) / 2;
    }

    void afficher() const {
        cout << "Étudiant: " << nom << "\nMoyenne: " << calc_moy() << endl;
    }
};

int main() {
    string nom;
    float note1, note2; // Utilisez float au lieu de int pour les notes.

    cout << "Entrez le nom: ";
    cin >> nom;

    cout << "Entrez la première note: ";
    cin >> note1;

    cout << "Entrez la deuxième note: ";
    cin >> note2;

    Etudiant E(nom, note1, note2);
    E.afficher();

    system("PAUSE");
    return 0;
}
```

Exercice 03 :

Écrivez un programme en C++ qui définit une classe appelée *Forme* avec un constructeur qui donne de la valeur à la largeur(x) et à la hauteur(y). Définir la méthode *aire()* dans les deux sous-classes *Triangle* et *Rectangle*, qui calculent l'aire. Dans la méthode principale *main*, définissez deux variables, un triangle et un rectangle, puis appelez la fonction *aire()* dans ces deux variables.

Notez que:

l'aire du triangle est = largeur * hauteur / 2

l'aire du rectangle est = largeur * hauteur.

Solution :

```
#include <iostream>
using namespace std;
class Forme {
protected:
    float x, y;
public:
    Forme(float x, float y)
    {
        this->x = x;
        this->y = y;
    }
};

class Rectangle: public Forme
{
public:
    Rectangle(float x, float y) : Forme (x, y) {}

    float aire()
    {
        return (x * y);
    }
};

class Triangle: public Forme
{
public:
    Triangle(float x, float y): Forme (x, y) {}

    float aire()
    {
        return (x * y / 2);
    }
};

int main (){
    Rectangle rectangle(2,3);
    Triangle triangle(2,3);
    cout << rectangle.aire() << endl;
```

```
    cout << triangle.aire() << endl;
    system("PAUSE");
    return 0;
}
```