Class Methods

Methods are **functions** that belongs to the class.

There are two ways to define functions that belongs to a class:

- **Inside class definition**
- **Outside class definition**

Outside class definition

To define a function outside the class definition, **you have to declare it inside the class and then define it outside of the class**.

This is done by **specifiying the name of the class, followed the scope resolution :: operator, followed by the name of the function**:

Example (1)Inside class definition

```
class Retangler
{
public:
    int height, width;

    int Area()
    {
        return height * width;
    }
};
```

```
int main()
{    Retangler ob;
    cout << "Enter the height\n ";
    cin >>ob.height ;
    cout << "Enter the width \n";
    cin >> ob.width;
    cout << ob.Area();
}
```

Example (2)Outsaid class definition

```
class Rectangler {
public:
    int height, width;

    int Area();
};
int Rectangler :: Area()
{
    return height * width;
}
```

```
int main()
{    Retangler ob;
    cout << "Enter the height\n ";
    cin >>ob.height ;
    cout << "Enter the width \n";
    cin >> ob.width;
    cout << ob.Area();
}
```

Example (3)Inside class definition

```
class Retangler {
private:
    int height,width;
public:
    void set(int h, int w)
    {
        height = h;
        width = w;
     }
    int get()
    {
        return height * width;
    }
};
```

```
int main()
{
            Retangler ob;
           int x, y;
           cout << "Enter No1\n ";
           cin >> x;
           cout << "Enter No2 \n";
                    cin >> y;
           ob.set(x, y);
           cout<<ob.get();
}
```

Example (4)Outsaid class definition

```
class Retangler {
private:
   int height, width;
public:
   void set(int h, int w);
   int get();

};
void Retangler::set(int h, int w)
{
   height = h;
   width = w;
  }
int Retangler::get()
{
   return height * width;
}
```

```
int main()
{
   Retangler ob;
   int x, y;
   cout << "Enter No1\n ";
   cin >> x;
   cout << "Enter No2 \n";
   cin >> y;
   ob.set(x, y);
   cout << ob.get();
}
```

```
class Sum {
public:
    int x, y;

     int Result()
    {
        return x + y;
    }
  };

int main()
{
    Sum ob;
    cout << "Enter No1\n ";
    cin >> ob.x;
    cout << "Enter No2 \n";
    cin >> ob.y;
    cout << ob.Result();
     }
```

Function Inside Calss

```
class Sum {
public:
    int x, y;
    int Result();
  };
int Sum::Result()
{
    return x + y;
}

int main()
{
    Sum ob;
    cout << "Enter No1\n ";
    cin >> ob.x;
    cout << "Enter No2 \n";
    cin >> ob.y;
    cout << ob.Result();
}
```

Function Outside Calss

Function Inside Calss

```
class Employees
{
private:
  string FirstName, LastName;
  int Age;
public:
  void setEmp(string Name1,string Name2, int a)
  {
     FirstName=Name1;
     LastName = Name2;
     Age = a;
          }
void getEmp( )
{  cout << "FirsName= " << FirstName << endl;
   cout << "LastName= " << LastName << endl;
   cout << "Age= " << Age << endl;
          }
};
```

```
int main( )
{
  Employees ob;
  string name1, name2;
  int AgeNo;
    cout << "Enter FirsName \n ";
          cin >> name1;
   cout << "Enter LastName \n";
          cin >> name2;
  cout << "Enter Age \n";
          cin >> AgeNo;

     ob.setEmp(name1, name2, AgeNo);

     ob.getEmp( );
}
```

Function Outside Calss

```
class Employees
{
private:
    string FirstName, LastName;
    int Age;
public:
    void setEmp(string Name1, string Name2, int a);
    void getEmp();
   };
void  Employees :: setEmp(string Name1, string Name2, int a)
{
    FirstName = Name1;
    LastName = Name2;
    Age = a;
}
void Employees :: getEmp()
{    cout << "FirsName= " << FirstName << endl;
    cout << "LastName= " << LastName << endl;
    cout << "Age= " << Age << endl; }
```

```
int main( )
{
  Employees ob;
  string name1, name2;
  int AgeNo;
    cout << "Enter FirsName \n ";
            cin >> name1;
   cout << "Enter LastName \n";
            cin >> name2;
  cout << "Enter Age \n";
            cin >> AgeNo;

      ob.setEmp(name1, name2, AgeNo);

      ob.getEmp( );
}
```