# Object-Oriented Programming (oop)


By

## Dr. Rabih Ali Sbera

**Department of Computer Engineering Techniques -2nd**


**2025-2026**

Let's learn programming




lets learn online

**Member Function**

**Member Functions**

- **Member functions** are functions **declared inside a class**.
- They can directly access **private, protected, and public members** of the class.
- They are called using an object, pointer, or reference to the class.

**Example:**

```
#include <iostream>
using namespace std;
class Rectangle {
private:
   int length;
   int width;
public:
   void setValues(int l, int w) { // Member function
      length = l;
      width = w; }
   int area() { // Member function
      return length * width;
   }
   void printArea() { // Member function
      cout << "Area: " << area() << endl; }};
int main() {
   Rectangle r;
   r.setValues(5, 4);   // Calling member functions
   r.printArea();}
```

**Area: 20**

## Non-Member Functions

- **Non-member functions** are declared **outside the class**.
- They **cannot directly access private members** of the class unless they are declared as friend or use public getters.
- They are called **directly**, not through an object method call.

## Example:

```
#include <iostream>
using namespace std;
class Rectangle {
private:
   int length;
   int width;
public:
   void setValues(int l, int w) {
      length = l;
      width = w;    }
   int getLength() { return length; }
   int getWidth() { return width; }};
// Non-member function
int computeArea(Rectangle r) {
   return r.getLength() * r.getWidth(); // Uses member functions to access data
}
int main() {
   Rectangle r;
   r.setValues(5, 4);
   cout << "Area using non-member function: " << computeArea(r) << endl; }
```

**Area using non-member function: 20**