Programming Technique II – SCJ1023

Introduction to Class

Associate Prof. Dr. Norazah Yusof

What is a object oriented programming?

- Object-oriented programming is a method of writing software that centered on the object.
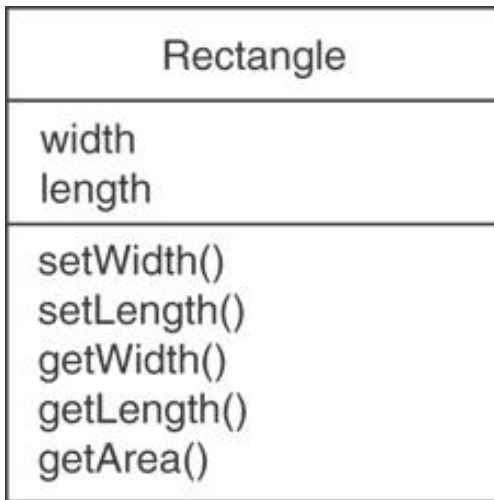
What is a class?

- Class is a description of an object, consists of attributes and functions that are members related to the class.

- Attributes are member variables of a class

- Functions are behaviors of a class

- Object is an instance of a class.

What is a Class diagram?

- Class diagram is a diagrams for graphically depicting object-oriented systems

- A class diagram has three main sections:
  - Class name
  - Member variables
  - Member functions

Class name goes here ⟶

Member variables are listed here ⟶

Member functions are listed here ⟶
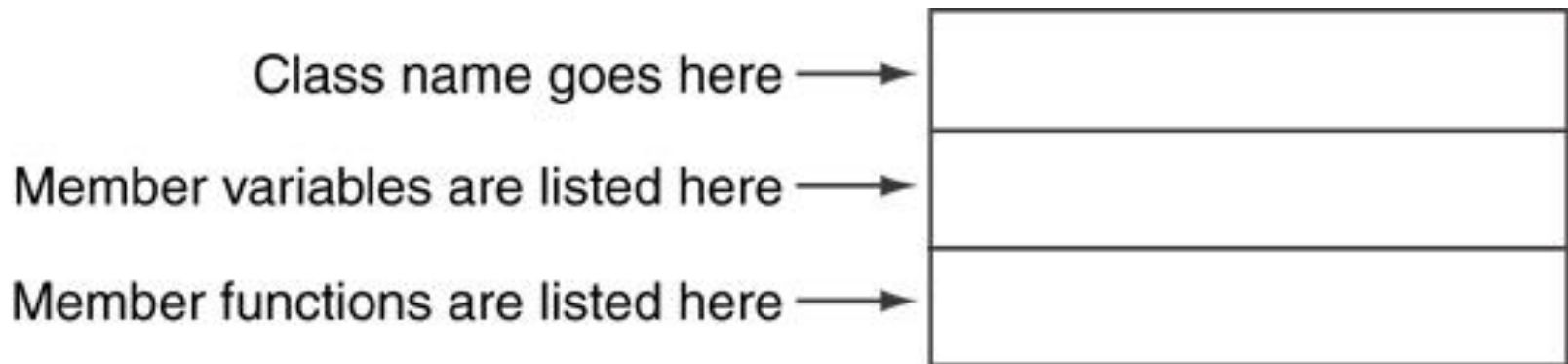
Example of class diagram : A Rectangle class

Rectangle
width length
setWidth() setLength() getWidth() getLength() getArea()

```cpp
class Rectangle
{
    private:
        double width;
        double length;
    public:
        bool setWidth(double);
        bool setLength(double);
        double getWidth() const;
        double getLength() const;
        double getArea() const;
};
```

What is class access specifiers?

- Class access specifiers are keywords used in class declaration to specify how class members may be accessed.
  - **public**: functions outside the class can access the attributes and functions
  - **private**: only functions that are members of the class can access the attributes and functions

Access specifier in C++

- Can be listed in any order in a class

- Can appear multiple times in a class

- If not specified, the default is `private`

Example of class definition

```
class Rectangle
{
    private:
        double width;
        double length;
    public:
        void setWidth(double);
        void setLength(double);
        double getWidth() const;
        double getLength() const;
        double getArea() const;
};
```

Member Attributes

Member functions

Access specifier

ocw.utm.my
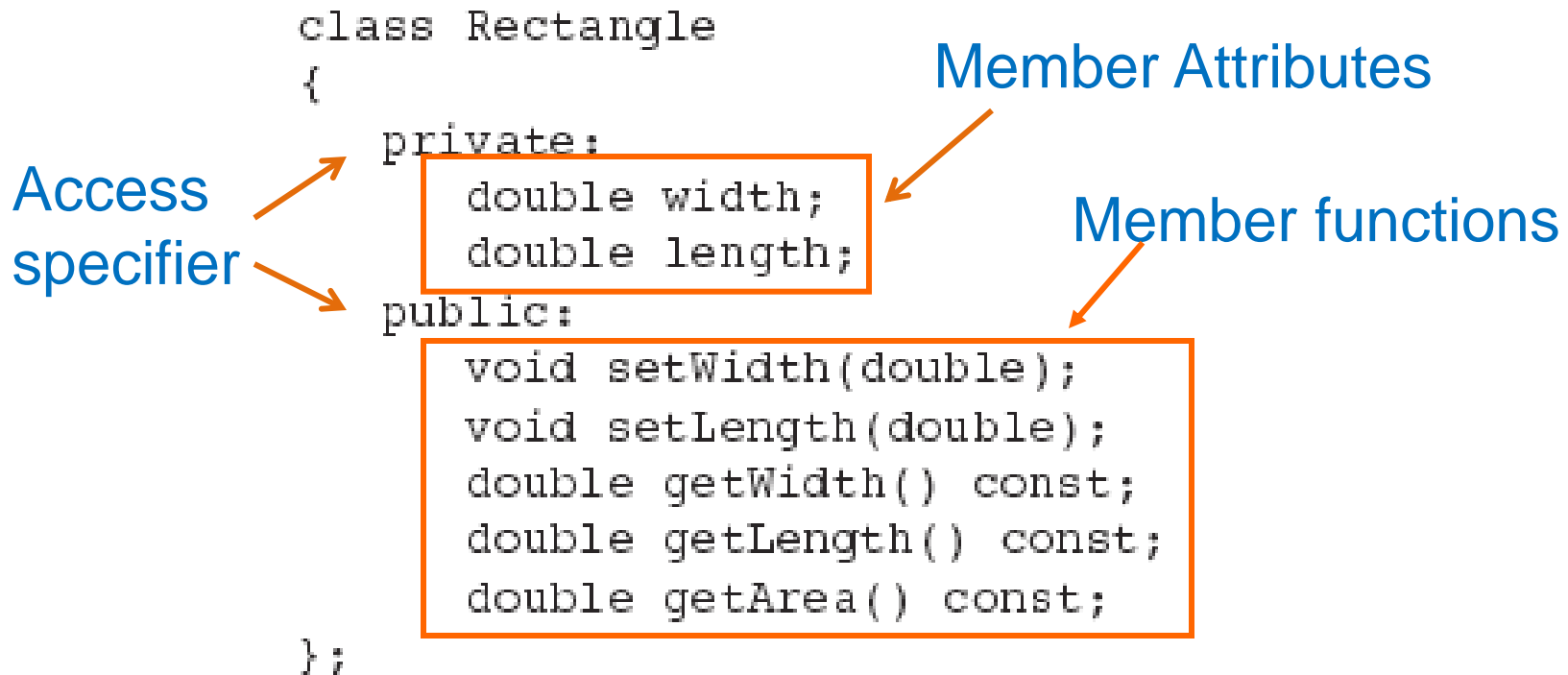
What is accessor function?

- Accessor is a public member function that retrieves values from `private` member variables

- It is recommended to named the accessor function with the verb "set".

- This kind of member function do not change an object's data if it is marked `"const"`.

What is mutator function?

- Mutator is a member function that stores or changes values in private member variables.

- It is recommended to named the mutator function with the verb "get".

Instance of a class

- An object is an instance of a class
- Use the dot operator to access members:

```
r.setWidth(5.2);
cout << r.getWidth();
```

- Cannot access `private` member using dot operator (Compiler error)

What is inline member function?

- Inline Member Function is when a member function declaration is written inside a class declaration to replace member function definition.

Pointer to an object

- Can define a pointer to an object:

  ```
  Rectangle *rPtr;
  ```

- Can access public members via pointer:

  ```
  rPtr = &otherRectangle;
  rPtr->setLength(12.5);
  cout << rPtr->getLenght() << endl;
  ```

Dynamically allocating an object

- We can also use a pointer to dynamically allocate an object.

```
1   // Define a Rectangle pointer.
2   Rectangle *rectPtr;
3
4   // Dynamically allocate a Rectangle object.
5   rectPtr = new Rectangle;
6
7   // Store values in the object's width and length.
8   rectPtr->setWidth(10.0);
9   rectPtr->setLength(15.0);
10
11  // Delete the object from memory.
12  delete rectPtr;
13  rectPtr = 0;
```

Constructors

- Member function that is automatically called when an object is created

- Useful for initializing member variables or performing other setup operations.

- Constructor function name is class name

- Has no return type

- Format:

  ClassName :: ClassName (ParameterList)

Default Constructors

- A default constructor is a constructor without any argument.

- C++ will prepare a default constructor if there is no constructor defined by the constructor.

- A simple instantiation of a class (with no arguments) calls the default constructor:

```
Rectangle r;
```

What is a Destructor?

- Destructor is a member function that will be invoked automatically when an object is destroyed

- Destructor is named by its class with a tilde at the front such as `~Rectangle`

- Destructor cannot be overloaded and there is only one destructor per class

- It has no arguments and has no return type.

- Destructor will release the dynamic memory, if the constructor allocates it.

What is overloading constructors?

- Overloading constructor is a class with more than one constructor, but must have different parameter lists:

```
Rectangle();
Rectangle(double);
Rectangle(double, double);
```