Constructors

Initializing object data is the responsibility of the constructor, which always has **the same name as the class** and which **never has a return type**.

```
class Time
{
public:
    Time();      // default constructor
    ...
};
```

A constructor is a member function which **initializes an object into a well-formed state** before clients start manipulating it. When C++ creates an object from a class:

1. It **allocates a block of memory** large enough to store the data elements.
2. It passes **the address of that block** of memory to the constructor function. The address is the this pointer inside the constructor function.

The constructor **is called automatically** whenever an object is created. If you have a class that defines a constructor, that constructor is **guaranteed to execute** whenever you create an object of the class type.

Default Constructors

The **default constructor** is the constructor which takes **no arguments** and which should initialize **all of its data members** to an appropriate **default** value. Alternatively, since C++11, you may **provide an initial value** when defining the data members, just as in Java.

If you do not provide a constructor, the compiler will "write" one for you. This is called the **synthesized default constructor**. If you use in-definition initializers, then this is perfect.