# Operator OverLoading

Its is just like function overloading except this works on operators instead and helps redefine how a operator works on customs user-defined data-types.

## Book Definition:

In Object-Oriented Programming (OOP), operator overloading is a powerful feature of C++ that allows developers to redefine the behavior of existing operators for user-defined types (like classes and structures). This enables objects to interact in intuitive ways, much like built-in data types, improving readability and code reusability.

**Book Answer to What is Operator Overloading?**

Operator overloading means providing a new definition for a C++ operator to work with user-defined data types.

A screenshot of a computer program

AI-generated content may be incorrect.

While using Operator Overloading use/give functionality to that operator that resemble what it traditionally is used for, its a good convention to follow as it make your code readable ,clean and easily debuggable.

But you are not forced to do it soo....:)

A computer screen shot of a program

AI-generated content may be incorrect.

Remember it only works on operators already built in and no more.

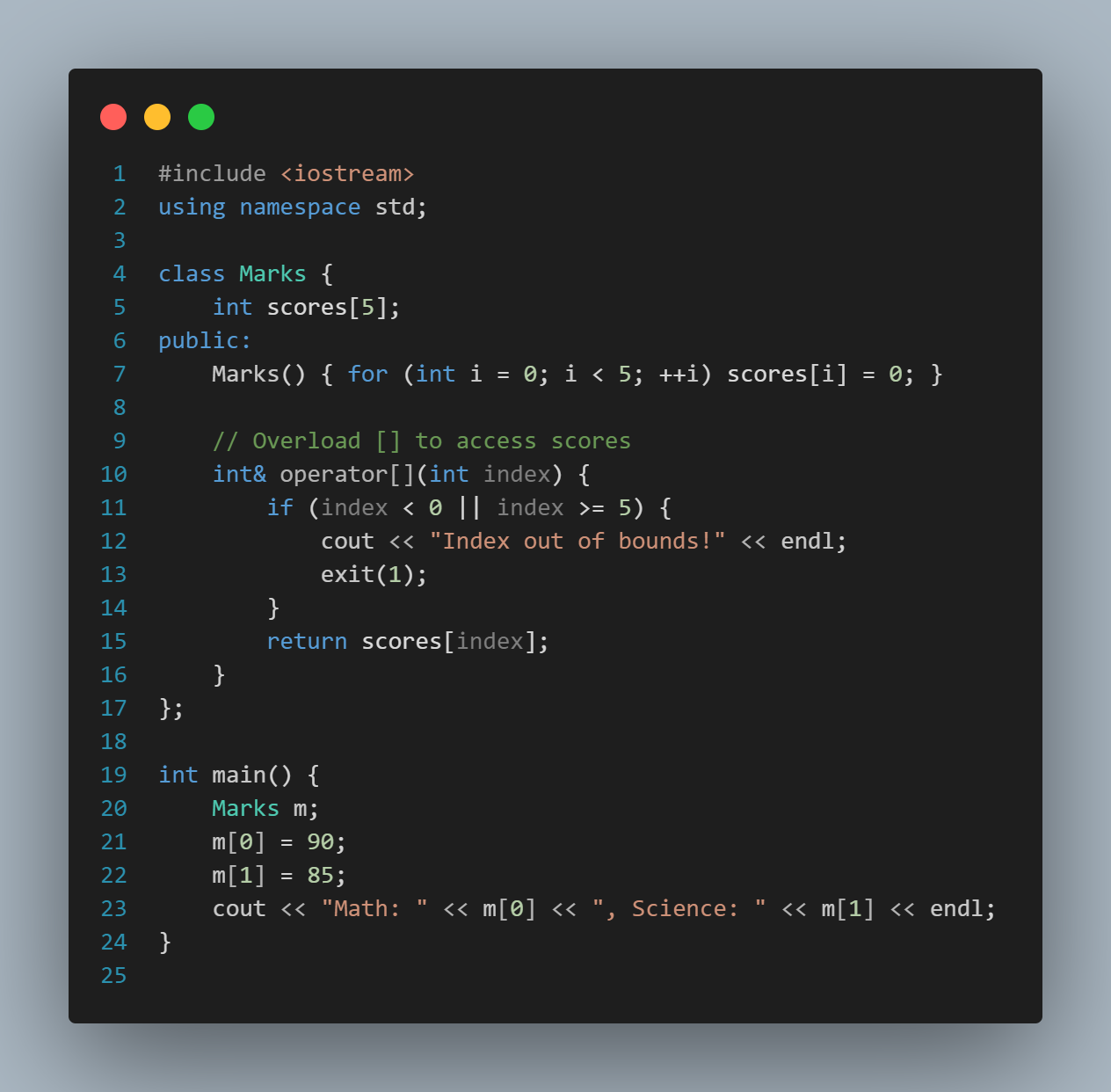
:: . .\* sizeof are non Overloadable Operators

While += -= + - = == \*= \* & ^ % + %= ! ++ -- << >> < > -> <= >= / /= | && || != [] () are Overloadable Operators

Use these to Verbose(short,readable,clean) the Code.

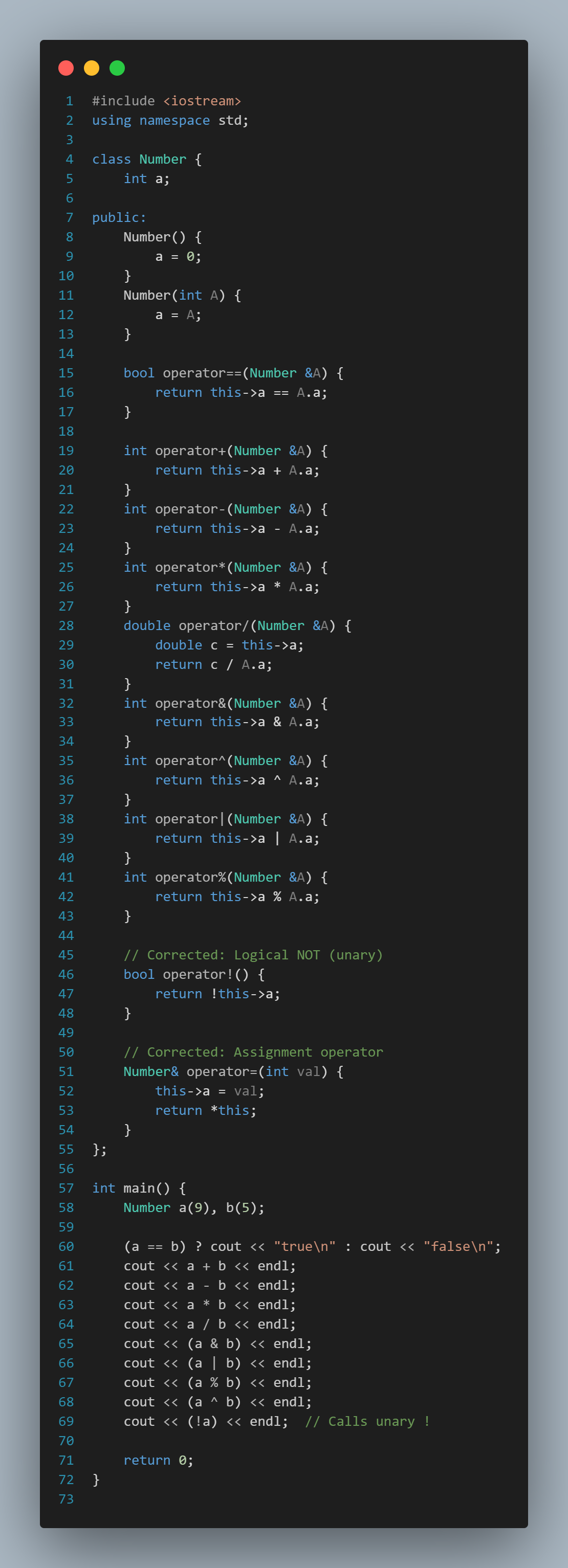
**Book Rules for Operator Overloading**

* You **cannot** create new operators.
* Some operators **cannot** be overloaded (e.g., ::, .\*, ., sizeof).
* At least one operand must be a **user-defined** type.
* Operator overloading should maintain the natural meaning of the operator to avoid confusion.



A screen shot of a computer

AI-generated content may be incorrect.



Now some examples ive taken from chatgpt

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.A computer screen shot of a program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

**Conclusion**

Operator overloading allows object-oriented programs in C++ to express actions between objects in a more intuitive and readable way. However, it should be used wisely—maintaining clarity and the natural meaning of operators. Mastery of operator overloading strengthens a programmer’s ability to build user-friendly and robust C++ applications.

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.

class ID {

int idNumber;

public:

ID(int id): idNumber(id) {}

bool operator==(const ID& other) {

return idNumber == other.idNumber;

}

};

A screenshot of a computer program

AI-generated content may be incorrect.

class Currency {

int rupees;

public:

Currency(int r = 0): rupees(r) {}

Currency operator+(const Currency& c) {

return Currency(rupees + c.rupees);

}

};