# Week 6: CopyConstructor

OBJECT ORIENTED PROGRAMMING (OOP)

# Learning Objectives

**After this lecture, students will be able to:**

1-Define the Copy Constructor

2-Understand when it is automatically called

3-Create a user-defined copy constructor

4-Explain shallow copy vs deep copy

5-Apply copy constructors in real examples

# What is a Copy Constructor

- A copy constructor is a special constructor that creates a new object as a copy of an existing object.

- ClassName(const ClassName &obj);

# When Is the Copy Constructor Called

- **The copy constructor is automatically invoked when:**
- **1-An object is initialized using another object.**
  - **ClassA obj2 = obj1;**
- **2-An object is passed by value to a function.**
- **3-A function returns an object.**

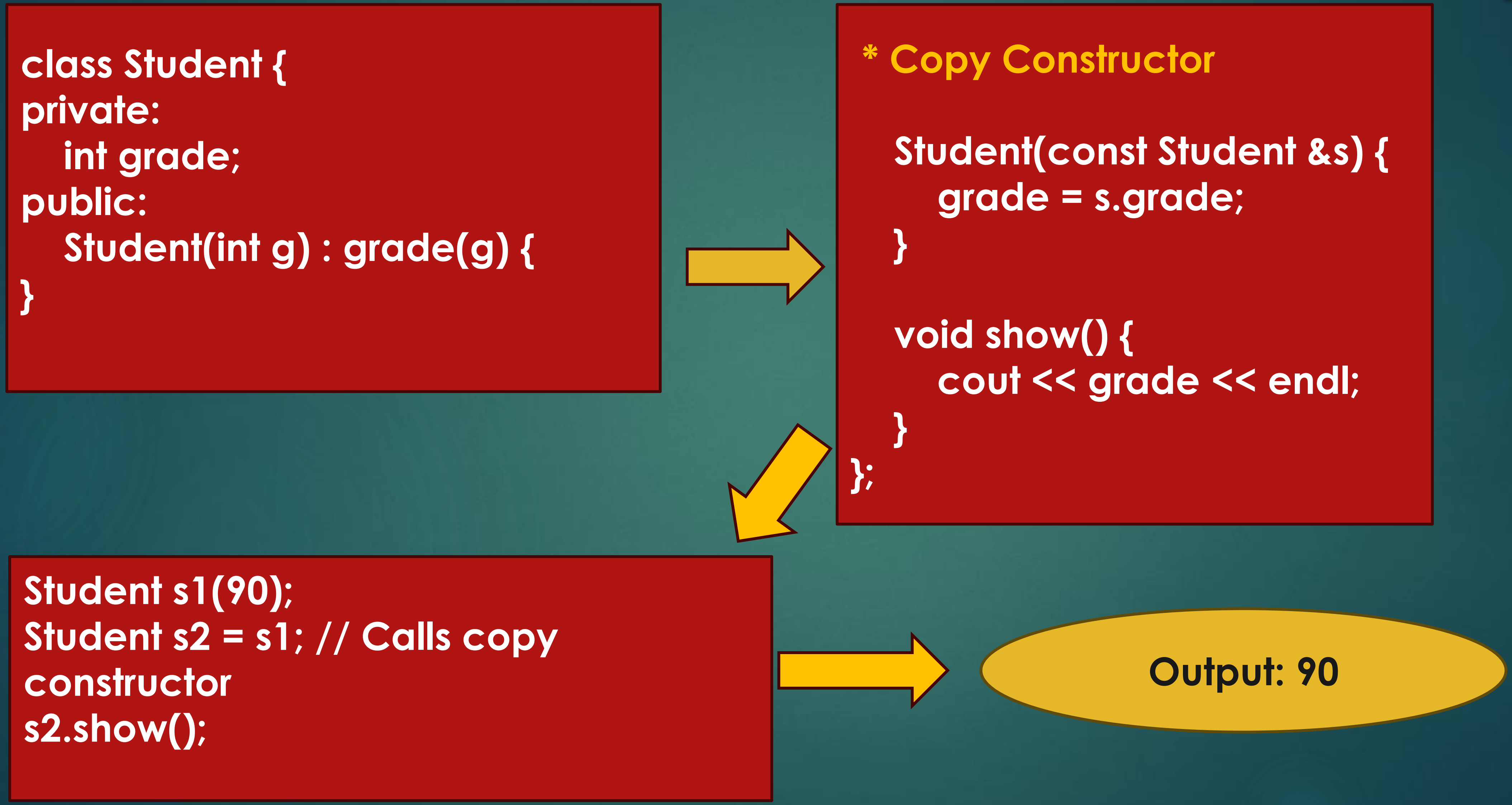
# Default Copy Constructor

- **If the programmer does not define a copy constructor, C++ provides a default one**

**1-Performs shallow copy**

**2- Copies all data members directly**

**3-Can cause issues with pointers and dynamic memory**



```
class Student {
private:
    int grade;
public:
    Student(int g) : grade(g) {
}
```

* Copy Constructor

```
    Student(const Student &s) {
        grade = s.grade;
    }

    void show() {
        cout << grade << endl;
    }
};
```

```
Student s1(90);
Student s2 = s1; // Calls copy
constructor
s2.show();
```

Output: 90