

C14 - Lab Experience: Copy Constructor and Copy Assignment in C++

Objective

The purpose of this lab is to understand and implement the **copy constructor** and **copy assignment operator** in C++.

- The **copy constructor** is used to create a new object as a copy of an existing object.
- The **copy assignment operator** (operator=) is used to copy values from one existing object to another that has already been initialized.

In this lab, you will implement and test these concepts in a **Person class**, where each object represents a person with a name, age, and a portrait file path.

Background

Copy Constructor

- A **copy constructor** is invoked when a new object is initialized using an existing object.
- Syntax: `Person(const Person& other);`
- This ensures that the new object gets a proper copy of the original, instead of just copying pointers and risking unintended shared state.

Copy Assignment Operator

- The **copy assignment operator** is invoked when an existing object is assigned another existing object (after initialization).
 - Syntax: `Person& operator=(const Person& other);`
 - This operator ensures deep copying when necessary to prevent shared memory issues.
-

Instructions

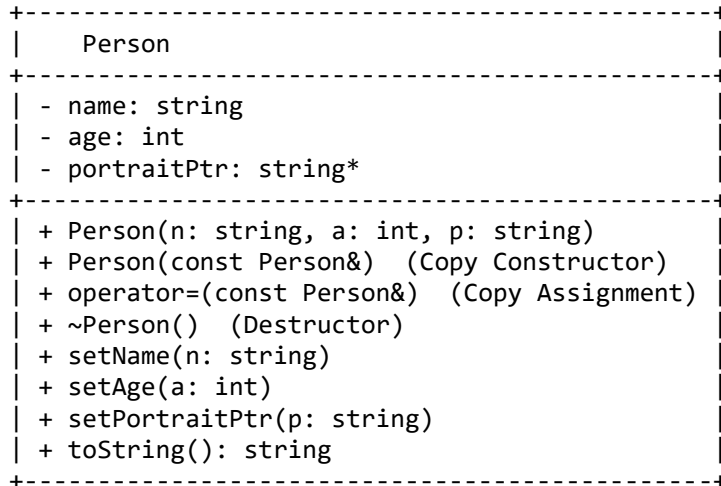
Step 1: Implement the Person Class

Modify Person.h and Person.cpp to include the following:

1. A **copy constructor** that properly copies data.
2. A **copy assignment operator** (operator=) to correctly assign one Person object to another.
3. Methods for setting and getting attributes:

- setName(string), setAge(int), setPortraitPtr(string).
- toString() that returns a formatted string representation of the object.

Class Diagram: Person class



Step 2: Implement the Main Program

Modify C14-Copy-Constructor-Person.cpp to do the following:

1. Create an instance of Person, p1.
 2. Use the **copy constructor** to create p2 from p1.
 3. Use the **copy assignment operator** to copy p1 into p3.
 4. Modify p2 and check whether p1 remains unaffected.
 5. Assign p2 to p3 and display all objects.
-

Expected Output

```

p1   Name: Homer, Age: 38, Portrait: c:/pictures/homer.jpg
p2   Name: Homer, Age: 38, Portrait: c:/pictures/homer.jpg
p3   Name: Homer, Age: 38, Portrait: c:/pictures/homer.jpg
p1   Name: Homer, Age: 38, Portrait: c:/pictures/homer.jpg
p2   Name: Homero, Age: 65, Portrait: c:/pictures/brad.jpg
p3   Name: Homero, Age: 65, Portrait: c:/pictures/brad.jpg

```

All done!