Tutorial 34 - Copy Constructor in C++

Key Concepts

1. **Copy Constructor**:
   - A special constructor that creates a copy of an existing object.
   - Takes a reference to the object being copied as a parameter.
   - Automatically supplied by the compiler if not explicitly defined.
2. **Usage**:
   - Helps create new objects with the same state as an existing object.
   - Explicitly invoked in certain scenarios like:
     - Passing objects by value.
     - Returning objects from functions.
     - Initializing a new object with an existing one.

Code Example and Explanation

Code Snippet 1: Copy Constructor Example

```
#include<iostream>
using namespace std;
class Number {
    int a;
public:
    Number() { a = 0; } // Default Constructor
    Number(int num) { a = num; } // Parameterized Constructor
    Number(Number &obj) { // Copy Constructor
        cout << "Copy constructor called!!!" << endl;
        a = obj.a;
    }
    void display() {
        cout << "The number for this object is " << a << endl;
    }
};

```

Explanation:

1. **Default Constructor**:
   - Initializes `a` to 0.
2. **Parameterized Constructor**:
   - Initializes `a` with the provided value.
3. **Copy Constructor**:
   - Copies the value of `a` from an existing object.
   - Prints a message to indicate invocation.

Code Snippet 2: Main Program

```
int main() {
    Number x, y, z(45), z2;
```

```
    x.display();
    y.display();
    z.display();
    Number z1(z); // Copy constructor invoked
    z1.display();
    z2 = z; // Copy constructor not called
    z2.display();
    Number z3 = z; // Copy constructor invoked
    z3.display();
    return 0;
}

```

Execution:

1. **Object Creation**:
   - `x`, `y`: Default constructor.
   - `z`: Parameterized constructor (value `45`).
2. **Copy Constructor Invocation**:
   - `z1(z)`: New object `z1` is created as a copy of `z`.
   - `z3 = z`: Invokes the copy constructor because a new object is being created.
3. **Direct Assignment**:
   - `z2 = z`: Does not invoke the copy constructor because `z2` is already created.

Output

```
The number for this object is 0
The number for this object is 0
The number for this object is 45
Copy constructor called!!!
The number for this object is 45
The number for this object is 45
Copy constructor called!!!
The number for this object is 45

```

Short Notes for Notebook

Copy Constructor

1. **Definition**:
   - Special constructor used to create a copy of an object.
   - Takes a reference to the object being copied as a parameter.
2. **Syntax**:

```
ClassName(const ClassName &obj) {
    // Copy object properties
}

```

3. **Key Points**:
   - Invoked automatically during:
     - Object initialization with another object (e.g., `ClassName obj2 = obj1;`).

    - Passing objects by value.
    - Returning objects from a function.
- Not invoked during direct assignment (`obj2 = obj1`).

Code Example

```
class Number {
    int a;
public:
    Number() { a = 0; }               // Default Constructor
    Number(int num) { a = num; }      // Parameterized Constructor
    Number(const Number &obj) {       // Copy Constructor
        cout << "Copy constructor called!!!" << endl;
        a = obj.a;
    }
    void display() {
        cout << "The number for this object is " << a << endl;
    }
};

```

Main Program

```
int main() {
    Number x, y, z(45), z2;
    Number z1(z); // Copy constructor invoked
    z2 = z;       // Copy constructor not called
    Number z3 = z; // Copy constructor invoked
    return 0;
}

```

Output

```
Copy constructor called!!!
Copy constructor called!!!
```