Tutorial 31 - Constructor Overloading In C++

Key Concepts

1. **Constructor Overloading**:
   - A class can have multiple constructors with the same name but different parameter lists.
   - The constructor executed depends on the number and types of arguments passed during object creation.
2. **Types of Constructors**:
   - **Default Constructor**: No parameters; assigns default values to data members.
   - **Parameterized Constructor**: Takes parameters to initialize data members.

Code Example and Explanation

Code Snippet 1: Constructor Overloading

```
#include <iostream>
using namespace std;
class Complex {
    int a, b;
public:
    // Default Constructor
    Complex() {
        a = 0;
        b = 0;
    }
    // Parameterized Constructor with 2 arguments
    Complex(int x, int y) {
        a = x;
        b = y;
    }
    // Parameterized Constructor with 1 argument
    Complex(int x) {
        a = x;
        b = 0;
    }
    void printNumber() {
        cout << "Your number is " << a << " + " << b << "i" << endl;
    }
};

```

Explanation:

1. **Class** `Complex`:
   - Private members: `a`, `b`.
   - Three constructors:
     - Default constructor: Assigns `0` to `a` and `b`.
     - Constructor with two parameters: Initializes `a` and `b` with passed values.
     - Constructor with one parameter: Initializes `a` with the passed value and `b` with `0`.
   - Function `printNumber()` prints the values of `a` and `b`.

Main Program (Code Snippet 2)

```
int main() {
    Complex c1(4, 6);  // Calls constructor with two parameters
    c1.printNumber();
    Complex c2(5);     // Calls constructor with one parameter
    c2.printNumber();
    Complex c3;        // Calls default constructor
    c3.printNumber();
    return 0;
}

```

Execution:

1. **Object** `c1`: Calls the constructor with two arguments `(4, 6)` and prints `Your number is 4 + 6i`.
2. **Object** `c2`: Calls the constructor with one argument `(5)` and prints `Your number is 5 + 0i`.
3. **Object** `c3`: Calls the default constructor and prints `Your number is 0 + 0i`.

Short Notes for Notebook

Constructor Overloading

1. **Definition**: A class having multiple constructors with the same name but different parameters.
2. **Key Points**:
   - Constructor executed depends on the arguments passed.
   - Types: Default and Parameterized constructors.

Code Example

```
class Complex {
    int a, b;
public:
    Complex() {           // Default Constructor
        a = 0;
        b = 0;
    }
    Complex(int x, int y) { // Parameterized Constructor (2 args)
        a = x;
        b = y;
    }
    Complex(int x) {       // Parameterized Constructor (1 arg)
        a = x;
        b = 0;
    }
    void printNumber() {
        cout << "Your number is " << a << " + " << b << "i" << endl;
    }
};

```

Usage in Main:

```
int main() {
    Complex c1(4, 6);  // Calls constructor with 2 arguments
    c1.printNumber();
```

```
    Complex c2(5);     // Calls constructor with 1 argument
    c2.printNumber();
    Complex c3;        // Calls default constructor
    c3.printNumber();
}

```

Output

```
Your number is 4 + 6i
Your number is 5 + 0i
Your number is 0 + 0i
```