Tutorial 29 - Constructors in C++

Key Points About Constructors

- **Definition**: A constructor is a special member function with the same name as the class, used to initialize objects.
- **Features**:
  a. No return type (not even `void`).
  b. Automatically invoked when an object is created.
  c. Can have default arguments.
  d. Cannot refer to their address.
  e. Typically declared in the **public** section of a class.

Code Example

Code Snippet 1: Constructor Example

```
#include <iostream>
using namespace std;
class Complex {
    int a, b;
public:
    Complex(void); // Constructor declaration
    void printNumber() {
        cout << "Your number is " << a << " + " << b << "i" << endl;
    }
};
// Constructor Definition
Complex::Complex(void) {
    a = 10;
    b = 0;
}

```

Explanation:

1. **Class** `Complex`:
   - Contains private members `a` and `b`.
   - Public methods:
     - Constructor initializes `a` and `b`.
     - `printNumber()` prints their values.
2. **Default Constructor**:
   - A constructor that takes no parameters.
   - Automatically assigns default values to `a` and `b` whenever an object is created.

Code Snippet 2: Main Program

```
int main() {
    Complex c1, c2, c3;
    c1.printNumber();
    c2.printNumber();
```

```
    c3.printNumber();
    return 0;
}

```

Explanation:

1. Objects `c1`, `c2`, and `c3` are created.
   - For each object, the constructor is invoked automatically to initialize values.
2. The `printNumber()` function is called for each object to display the initialized values.

Output

```
Your number is 10 + 0i
Your number is 10 + 0i
Your number is 10 + 0i

```

Notes for Your Notebook

Constructors in C++

1. **Definition**:
   - A constructor is a special function with the same name as the class.
   - It initializes objects and is invoked automatically when an object is created.
2. **Characteristics**:
   - No return type.
   - Declared in the public section.
   - Can have default arguments.
   - Cannot refer to their address.
3. **Default Constructor**:
   - Takes no parameters.
   - Used to assign default values to class members.
4. **Code Example**:

```
class Complex {
    int a, b;
public:
    Complex(void); // Default constructor
    void printNumber();
};
Complex::Complex(void) {
    a = 10;
    b = 0;
}

```

5. **Usage in Main Program**:
   - When objects are created, the constructor is invoked for each.
   - Example:

```
Complex c1, c2, c3;
c1.printNumber();
```

```
c2.printNumber();
c3.printNumber();

```

- Output:

```
Your number is 10 + 0i
Your number is 10 + 0i
Your number is 10 + 0i
```