

# Tutorial 32 - Constructors With Default Arguments In C++

## Key Concepts

### 1. Default Arguments:

- Arguments provided in the constructor declaration are called default arguments.
- If a value is not passed for these arguments during object creation, the constructor automatically assigns the default values.

### 2. Constructor Declaration:

- Default values for parameters are specified when the constructor is declared.
- Example: `Simple(int a, int b=9, int c=8);`

---

## Code Example and Explanation

### Code Snippet 1: Constructor with Default Arguments

```
1 #include<iostream>
2 using namespace std;
3 class Simple {
4     int data1, data2, data3;
5 public:
6     // Constructor with Default Arguments
7     Simple(int a, int b = 9, int c = 8) {
8         data1 = a;
9         data2 = b;
10        data3 = c;
11    }
12    void printData();
13 };
14 void Simple::printData() {
15     cout << "The value of data1, data2, and data3 is "
16         << data1 << ", " << data2 << " and " << data3 << endl;
17 }
18
```

### Explanation:

#### 1. Class Simple:

- Private members: `data1`, `data2`, `data3`.
- Constructor:
  - Accepts three parameters: `a`, `b`, and `c`.
  - Default values: `b = 9`, `c = 8`.
  - If `b` or `c` is not provided during object creation, their default values are used.
- Function `printData()` prints the values of `data1`, `data2`, and `data3`.

---

### Main Program (Code Snippet 2)

```
1 int main() {
2     Simple s(12, 13); // Passes values for `a` and `b`; `c` uses default value
3     s.printData();
4     return 0;
}
```

```
5 }  
6
```

#### Execution:

1. **Object s :**
    - Constructor is called with `a = 12` and `b = 13`.
    - Since no value is provided for `c`, the default value `c = 8` is used.
  2. **Function printData() :**
    - Prints the values: `data1 = 12`, `data2 = 13`, and `data3 = 8`.
- 

## Short Notes for Notebook

### Constructors with Default Arguments

1. **Definition:**
    - Constructors with predefined default values for parameters.
    - If no value is passed for a parameter, its default value is used.
  2. **Key Points:**
    - Default values are specified during constructor declaration.
    - If all arguments are provided during object creation, default values are ignored.
- 

### Code Example

```
1 class Simple {  
2     int data1, data2, data3;  
3 public:  
4     Simple(int a, int b = 9, int c = 8) { // Default arguments  
5         data1 = a;  
6         data2 = b;  
7         data3 = c;  
8     }  
9     void printData() {  
10        cout << "The value of data1, data2, and data3 is "  
11            << data1 << ", " << data2 << " and " << data3 << endl;  
12    }  
13 };  
14
```

### Usage in Main:

```
1 int main() {  
2     Simple s(12, 13); // `c` uses default value (8)  
3     s.printData();    // Prints: 12, 13, 8  
4     return 0;  
5 }  
6
```

### Output

```
1 The value of data1, data2, and data3 is 12, 13 and 8
```