

## Experiment No. 6

Date :

Aim :

Write a C++ program to implement the concept of default constructor, Parameterized constructor and copy constructor in C++.

Date:

**Aim:** Write a C++ program to implement the concept of default constructor, Parameterized constructor & copy constructor in C++.

**Theory:** Constructor in C++ is a special method that is invoked automatically at an time of an object of a class is created. It is used to initiate the data members of new objects generally. The constructor in C++ has the same name as the class or a class or structure. It constructs the value i.e. provides data for the object which is why it is known as a constructor.

**Characteristics of Constructor in C++:**

- i) The name of the constructor is the same as it's class name.
- ii) Constructor are mostly declared in the public section of the class though they can be declared in the private section of class.
- iii) Constructor do not return values, hence they do not have a return type.
- iv) A constructor gets called automatically when we create the object of the class.

## Types of Constructor:

There are 3 types of constructors in C++:

i) Default Constructor: No parameters. They are used to create object with default values.

ii) Parameterized constructor: Takes parameters used to create an object with specific method's initial values.

iii) Copy constructor: Takes a reference to another object of the same class. Used to create a copy of an object.

### Syntax of Default Constructor:

```
class_name() {
    // body of constructor
}
```

### Syntax of Parameterized constructor:

```
class_name(parameters....) {
    // body of constructor
}
```



Syntax of copy constructor :

```
class_name (class_name &obj_name) {
    // body containing logic
}
```

Code : Default Constructor :

```
#include <iostream>
using namespace std;

class SimpleClass {
public:
    SimpleClass () {
        cout << "Default constructor
        called" << endl;
    }
};

int main () {
    SimpleClass obj;
    return 0;
}
```

Output: Default constructor called.

Code: Parameterized Constructor:

```
#include <iostream>
```

```
using namespace std;
```

```
class SimpleClass {
```

```
    int number;
```

```
public:
```

```
    SimpleClass (int num) {
```

```
        number = num;
```

```
        cout << "Parameterized constructor
```

```
called, Number : "<< number <<
```

```
endl;
```

```
    }
```

```
};
```

```
int main () {
```

```
    SimpleClass obj (42);
```

```
    return 0;
```

```
}
```

Output: Parameterized Constructor called, Number : 42

Code :

Copy Constructor :

#include &lt;iostream&gt;

using namespace std;

class SimpleClass {

int number;

public:

SimpleClass (int num) {

number = num;

cout << "Parameterized  
constructor called, Number  
:" << number << endl;

}

SimpleClass (const SimpleClass &amp;obj) {

number = obj.number;

cout << "Copy constructor  
called, Copied number :"  
<< number << endl;

}

};

int main () {

SimpleClass obj1 (42);

SimpleClass obj2 = obj1;

return 0;

}



Conclusion: We have successfully implemented the concept of default constructor, parameterized constructor & copy constructor in C++.

```

#include <iostream>
using namespace std;

class Simple {
public:
    Simple() { cout << "Default constructor called\n"; }
    Simple(int i) { cout << "Parameterized constructor called\n"; }
    Simple(const Simple& s) { cout << "Copy constructor called\n"; }
};

int main() {
    Simple s1;
    Simple s2(10);
    Simple s3(s1);
    return 0;
}

```

Output: Parameterized Constructor called, Number : 42  
Copy Constructor called, Number : 42

Conclusion: We have successfully implemented the concept of Default constructor, parameterized constructor and Copy constructor in C++.

