

### 3.1 Concept of Constructor –

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

- Constructor is a member function of a class, whose name is same as the class name.
- Constructor is a special type of member function that is used to initialize the data members for an object of a class automatically, when an object of the same class is created.
- Constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.
- Constructor do not return value, hence they do not have a return type.

The prototype of the constructor looks like

<Class-name> (list-of-parameters);

Constructor can be defined inside the class declaration or outside the class declaration

- a. Syntax for defining the constructor within the class

```
<Class-name>(list-of-parameters)
{
    //constructor definition
}
```

- b. Syntax for defining the constructor outside the class

```
<class-name>: :<class-name>(list-of-parameters)
{
    //constructor definition
}
```

To create a constructor, use the same name as the class, followed by parentheses ():

#### Example

```
class MyClass {    // The class
public:            // Access specifier
    MyClass() {    // Constructor
        cout << "Hello World!";
    }
};

int main() {
    MyClass myObj; // Create an object of MyClass (this will call the constructor)
    return 0;
}
```

### 3.2Types of Constructor: Default Constructor, Parameterized Constructor

#### C++ Default Constructor

A constructor which has no argument is known as default constructor. It is invoked at the time of creating object.

Let's see the simple example of C++ default Constructor.

```
#include <iostream>
using namespace std;
class Employee
{
    public:
        Employee()
        {
            cout<<"Default Constructor Invoked"<<endl;
        }
};
int main(void)
{
    Employee e1; //creating an object of Employee
    Employee e2;
    return 0;
}
```

#### **Output:**

```
Default Constructor Invoked
Default Constructor Invoked
```

Example2:

```
#include<iostream>
using namespace std;
class student
{
    int rno;
    char name[50];
    double fee;
    public:
        student() // Explicit Default constructor
        {
            cout<<"Enter the RollNo:";
            cin>>rno;
```

```

        cout<<"Enter the Name:";
        cin>>name;
        cout<<"Enter the Fee:";
        cin>>fee;
    }

    void display()
    {
        cout<<endl<<no<<"\t"<<name<<"\t"<<fee;
    }
};

int main()
{
    student s;
    s.display();
    return 0;
}

```

## C++ Parameterized Constructor

A constructor which has parameters is called parameterized constructor. It is used to provide different values to distinct objects.

Let's see the simple example of C++ Parameterized Constructor.

```

#include <iostream>
using namespace std;
class Employee {
public:
    int id;//data member (also instance variable)
    string name;//data member(also instance variable)
    float salary;
    Employee(int i, string n, float s)
    {
        id = i;
        name = n;
        salary = s;
    }
    void display()
    {
        cout<<id<<" "<<name<<" "<<salary<<endl;
    }
};

```

```
int main(void) {  
    Employee e1 =Employee(101, "Sonoo", 890000); //creating an object of Employee  
    Employee e2=Employee(102, "Nakul", 59000);  
    e1.display();  
    e2.display();  
    return 0;  
}
```

**Output:**

```
101 Sonoo 890000  
102 Nakul 59000
```