



Q1. Can we have more than one constructor in a class?
 If yes, explain the need for such a situation.

Ans. Yes, we can have multiple constructors in a class.
 It means we can have combination of default, parameterized, etc. constructors.

eg:

```
class construct
```

```
{
```

```
    public:
```

```
    construct ()
```

```
    {
```

```
        cout << "This is constructor 1!" << endl;
```

```
    }
```

```
    construct (int a, int b)
```

```
    {
```

```
        cout << "Values passed are" << a << " "
```

```
        << b;
```

```
    }
```

```
};
```

```
int main ()
```

```
{
```

```
    cout << "\n Program for constructor: ";
```

```
    construct obj;
```

```
    construct obj1(10, 11);
```

```
    return 0;
```

```
}
```

//o/p:

Program for constructor

This is constructor 1

Values passed are 10 11

- Q1. Need for multiple constructors:-
- A class can have multiple constructors that assign the fields in different ways.
- Sometimes its beneficial to specify every aspect of an object's data by assigning parameters to the fields, but other times it might be appropriate to define only one or a few.

Q2. How is dynamic initialization of objects achieved?

- Ans. The dynamic initialization of objects means to initialize the data members of the class while creating the object.
- When we want to provide initial or default values to the data members while creating of object - we need to use dynamic initialization of objects.
 - It can be implemented by using parameterized constructors in C++.

eg:

```
class Student
{
private:
int rollno;
float per;
public:
Student (int r, float p)
{
rollno = r;
per = p;
}
}
```

```
void read ()
```

```
{
```

```
cout << "Enter roll number: ";
```

```
cin >> rollno;
```

```
cout << "Enter percentage: ";
```

```
cin >> perc;
```

```
}
```

```
void print ()
```

```
{
```

```
cout << endl;
```

```
cout << "Roll number: " << rollno << endl;
```

```
cout << "Percentage: " << perc << "% " << endl;
```

```
}
```

```
};
```

```
int main ()
```

```
{
```

```
int roll-no; float perc;
```

```
cout << "Enter roll number to initialize  
the object: ";
```

```
cin >> roll-no;
```

```
cout << "Enter percentage to initialize  
the object: ";
```

```
cin >> perc;
```

```
Student obj (roll-no, perc);
```

```
cout << "After initializing the object  
the values are: " << endl;
```

```
obj.print ();
```

```
obj.read ();
```

```
obj.print ();
```

```
return 0;
```

```
}
```

//o/p:

Enter roll number to initialize the object: 18
Enter percentage to initialize the object: 35
After initializing the object, the values are:

Roll number: 18
Percentage: 35%
Enter roll number: 69
Enter percentage: 69

Roll number: 69
Percentage: 69%

