

# C++ PROGRAMS

## Constructor and Destructor

### C++ program to illustrate the concept of Default Constructors

```
#include<iostream>
using namespace std;
class student
{
    public:
    student()
    {
        cout<< "Default Constructor";
    }
};
int main()
{
    student s;
    return 0;
}
```

### C++ program to illustrate the concept of Paramterized Constructors.

```
#include<iostream>
using namespace std;
class student
{
    public:
    int id,rn;
    student(int i ,int rn)
    {
        id=i;
        rn=r;
    }
    void show()
    {
        cout<< "Id is" << id;
        cout<< " Roll no is" << rn;
    }
};
int main()
{
    student s= student (1,101);
    s.show();
    return 0;
}
```

### C++ program to show the concept of Copy Constructors.

```
#include<iostream>
```

```

using namespace std;
class student
{
public:
    int id,rn;
    student(int i ,int rn)
    {
        id=i;
        rn=r;
    }
    student(student &stu)
    {
        id=stu.id;
        rn=stu.rollno;
    }
    void show()
    {
        cout<< "Id is" << id;
        cout<< " Roll no is" << rn;
    }
};
int main()
{
    student s= student (1,101);
    student s2=s;
    s.show();
    return 0;
}

```

## **C++ program to show the concept of Constructor Overloading .**

```

#include<iostream>
using namespace std;
class add
{
public:
    int c;
    add()    //Constructor with no arguments
    {
        c=0;
    }
    //Constructor with two arguments
    add(int a ,int b)
    {
        c=a+b;
    }
    void show()
    {
        cout<< c << endl;
    }
};
int main()
{
    add obj;
    add obj(10,20);
    obj.show();
    obj2.show();
    return 0;
}

```

```
}
```

## **C++ program to show the concept of Destructor**

```
#include<iostream>
using namespace std;
class student
{
    public:
    student()
    {
        cout<< "Constructor Called";
    }
    ~student()
    {
        cout<< "Destructor Called";
    }
};
int main()
{
    student s;
    return 0;
}
```