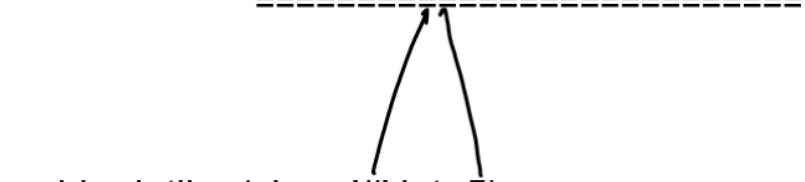
Default Function Arguments (DFA)



```
void printline(char='#',int=5);
int main()
{
    printline('A',10);
    printline('$');
    printline();
    getch();
    return 0;
}
```

printline('A',10); ⟶ printline('A',10);

printline('$'); ⟶ printline('$',5);

printline(); ⟶ printline('#',5);

```
void printline(char ch,int n)
{
   for(int i=1;i<=n;i++)
            cout<<ch<<"   ";
  cout<<endl;
}
```

Default Function Arguments (DFA)

==========

1. DFA is a new technique for arguments introduced by C++ language and not present in C language.

2. Using DFA a programmer has to declare and define only one function but it can be called with different number of arguments

3. For a programmer DFA is a easier alternate to function overloading.

4. This is because in case of function overloading we have to define multiple function with the same name and then we can call them in different ways but in DFA we define just one function but we can call it in multiple different ways.

Drawback or Restriction on DFA

==========

Although DFA is a useful technique but it has one major restriction. The restriction is that **Default arguments must always be Trailing argument**.

In simple words it means that if an argument of a function is set to some default value then

1. Either it should be the last argument

or

2. All the arguments after it must also be given default value.

✓ void show (int = 10, int = 20, int = 30);

✗ void show (int = 10, int, int = 30);

✓ void show (int , int = 10, int = 30);

void printline (char = '#', int = 5);

✓ printline ( );

✗ printline (, 20);

✓ printline (100);
but it will becom
printline ('d', 5);

```
#include <iostream.h>
#include <conio.h>
class Student
{
   int roll;
   char grade;
   float per;
public:
          Student(int=0,char=' ',float=0.0);
          void get();
          void show();
};
Student::Student(int r,char g,float p)
{
    roll=r;
    grade=g;
    per=p;
}
```

Student P(0,' ',0.0);

```
void Student::get()
{
  cout<<"Enter roll,grade and per:";
  cin>>roll>>grade>>per;
}
void Student::show()
{
  cout<<roll<<grade<<per;
}

int main()
{
    Student S(10,'A',78.9);
    Student P;
    P.get();
    S.show();
    P.show();
    getch();
    return 0;
}
```