Benefits Of Function Overloading
==========
1. The overhead of remembering function names is not on the programmer . Rather he just have to remember one single name and he can call diff versions of the same function.

2. Code looks more symmetrical.

When We Should Perform and When We Should Avoid Overloading ?
==========
1, We should always overload functions which perform same task but on diff arguments. For ex: suppose we want to swap two int and two floats. Since swapping logic is same so we must overload them as:

```
void swap(int *,int *);
void swap(float*,float *);
```

2. We should never overload functions which have diff tasks to do. For ex: suppose we want a function to calculate SQUARE and another function to calculate SQUARE ROOT . In this case we should avoid overloading since both the tasks are different.

Constructor Overloading

Just like we can overload functions , similalrly we also can overload constructors. That is , a single class can have multiple constructors.

But every constructor in the class must look different than other constructors of the same class w.r.t its arguments.

Moreover when the object of a class gets created then the compiler will not call every constructor present in the class. Rather it will call only one constructor which will be the BEST MATCH with arguments passed to object.

```
#include <iostream.h>
#include <string.h>
#include <conio.h>
class Emp
{
	int age;
	char name[20];
	float sal;
public:
	Emp();
	Emp(int,char *,float);
	void show();
};
 Emp::Emp()
{
  cout<<"Enter age,name and sal:";
  cin>>age>>name>>sal;
}
Emp::Emp(int a,char *p,float s)
{
	age=a;
	strcpy(name,p);
	sal=s;
}
void Emp::show()
{
	cout<<age<<","<<name<<","<<sal<<endl;
}
int main()
{
	clrscr();
	Emp E(21,"Rahul",25000.0);
	Emp F;
	E.show();
	F.show();
	getch();
	return 0;
}
```

Assignment:
=========
WAP to create a class called Box having 3 int data members for storing lenght, breadth and height of the Box.
Also provide following constructors and member function in the class:

1. A non parametrized constructor which should initialize length, breadth and height with user input.

2. A single parametrized constructor which should accept an int as argument and should initialize all data members with it.

3. A triple parametrized constructor which should initialize lenght, breadth and height with dif values passed as arg.

4. A member function called show() to displa l,b and h.
Finally design the function main() , create 3 objects of Box in such a way that every object calls a different constructor. At the end display their values.