# Assignment -3

### Default constructor

1.create defaule constructor and display following output(“this is default constructor”).

### Parameterized  constructor

2.find area of rectangle using parameterized constructor.

### Copy constructor

3. find area of rectangle using copy constructor.

## Operator overloading

4.write a c++ program to overload unary operator ++.

5. write a c++ program to overload unary operator --.

6. write a c++ program to overload binary operator +.

7. write a c++ program to overload binary operator -.

8. write a c++ program to overload binary operator *.

9. write a c++ program to overload binary operator /.

10. write a c++ program to overload relational operator = =.(codding is given below)

11. write a c++ program to overload relational operator + =.(codding is given below)

12. write a c++ program to overload relational operator - =.(codding is given below)

13. write a c++ program to overload relational operator ! =.(codding is given below)

## All are in one programming(all operator)

```
#include<iostream.h>
 #include<conio.h>
 class Test
{
public:
int a,b;
Test()

{
a=b=0;
}
int operator == (Test &t)
{
if(t.a==a)
{
return 0;
}
else
{
return 1;
}
}
int operator >= (Test &t)
{
if(t.a>=a)
{
return 0;
}
else
{
return 1;
}
}
int operator <= (Test &t)
{
if(t.a<=a)
{
return 0;
}
else
{
return 1;
}


}
int operator != (Test &t)
{
if(t.a!=a)
{
return 0;
}
else
{
return 1;
}
}
int operator += (Test &t)
{
t.a+=a; return t.a;
}
int operator -= (Test &t)
{
t.a-=a; return t.a;
}
int operator *= (Test &t)
{
t.a*=a; return t.a;
}
int operator &&(Test &t)
{
if(t.a&&a)
{
return 1;
}
else
{
return 0;
}
}
int operator || (Test &t)
{
if(t.a||a)
```

```
{
return 1;
}
else
{
return 0;
}


}
};
void main()
{

Test t1,t2,t3;

int num1,num2,num3,num4,num5,num6,num7,num8,num9,num10;
 clrscr();
t1.a=1;
 t2.a=0;

num1=(t1==t2);
num2=(t1>=t2);
 num3=(t1<=t2);
 num4=(t1!=t2);
num5=(t1+=t2);
 num6=(t1-=t2);
 num7=(t1*=t2);
num9=(t1&&t2);
 num10=(t1||t2);

cout<<endl<<"num=="<<num1;
 cout<<endl<<"num>="<<num2;
 cout<<endl<<"num<="<<num3;
cout<<endl<<"num!="<<num4;
 cout<<endl<<"num+="<<num5;
 cout<<endl<<"num-="<<num6;
 cout<<endl<<"num*="<<num7;
 cout<<endl<<"num&&"<<num9;
cout<<endl<<"num||"<<num10;
 getch();
}
```