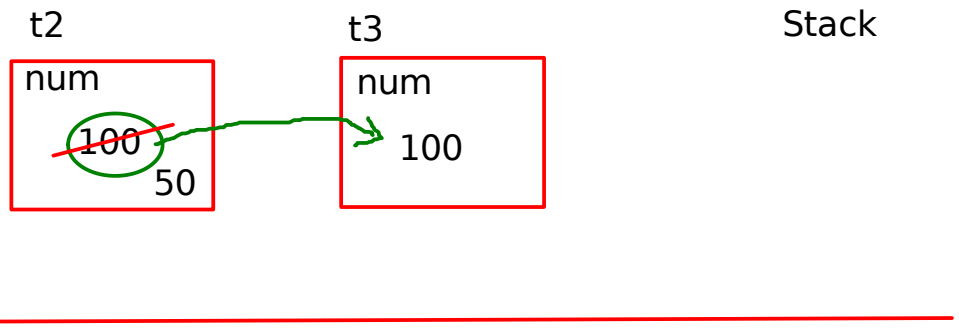
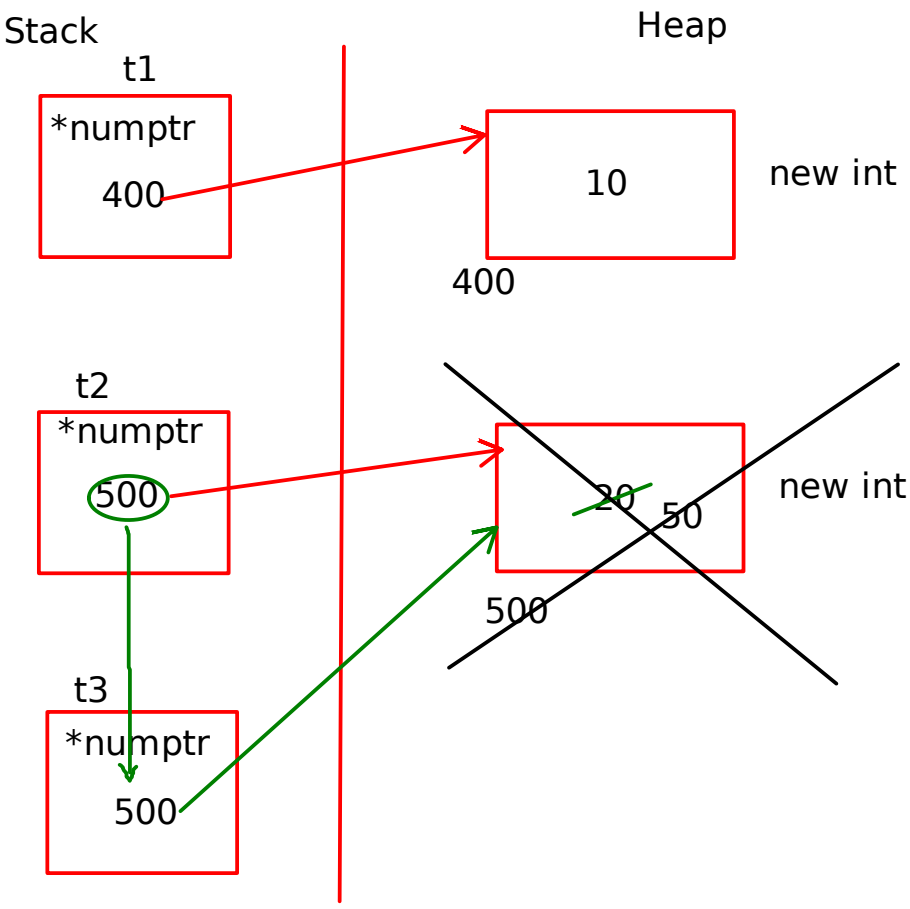
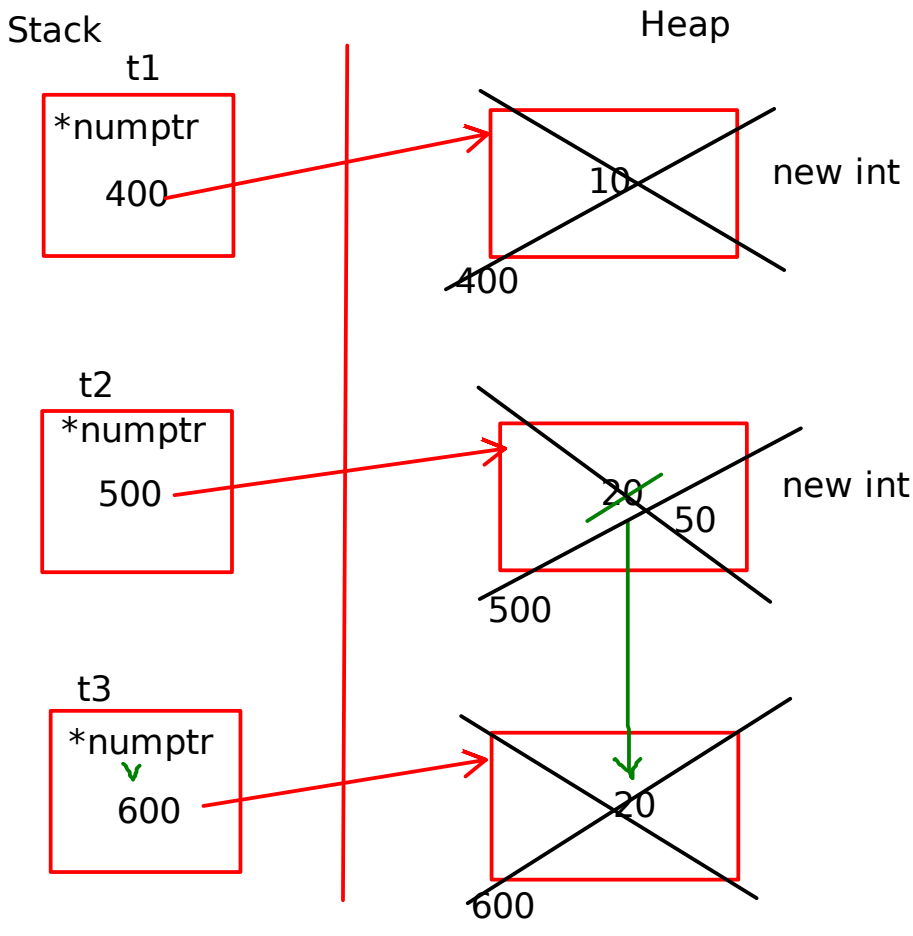
```
ifstream fin
ofstream fout

while(getline(fin,data))
{
cout<<data<<endl;
stringstream line(data);
}
```

class -> private / protected -> obj -> non member functions -> friend

```
   // copy ctor
   Test(Test &t){

   }
```







"sunbeam"+"infotech"

M1+M2

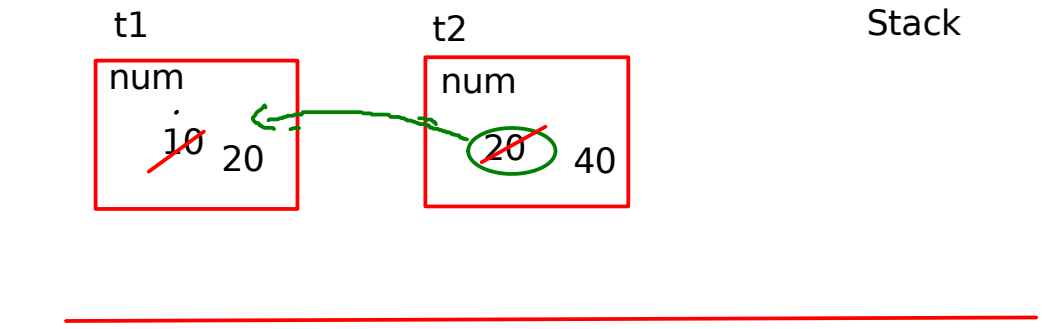
+ --> Obj + Obj
>> --> cin>>Obj
<< --> cout<<Obj

Why to do the operator overloading

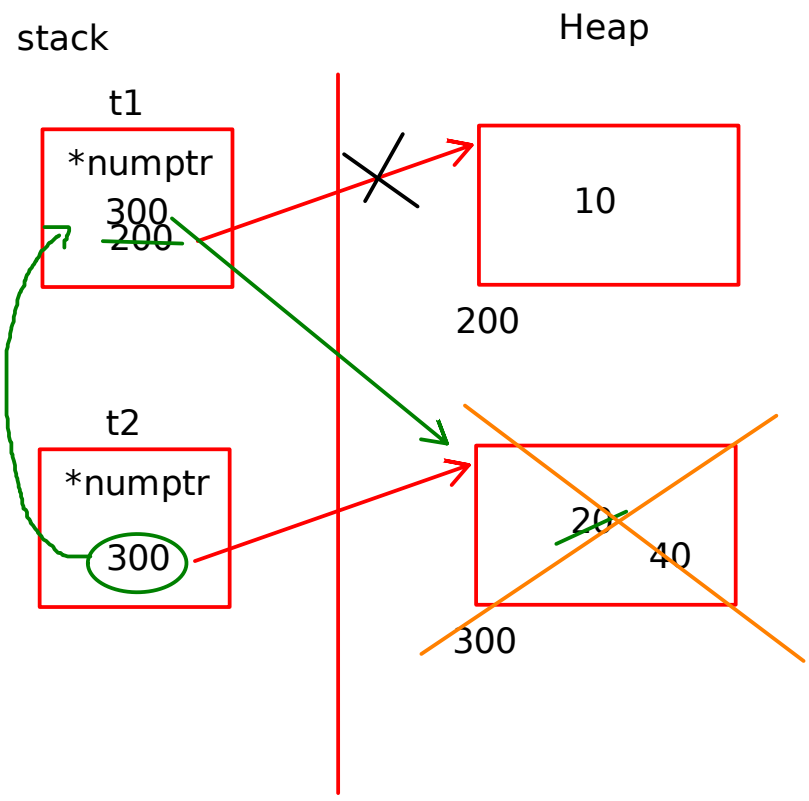
```
Matrix m1;
Matrix m2;
Matrix m3 = m1 + m2;

// withinh the class
m1.operator+(m2)

// global scope
operator+(m1,m2);
```

```
Student s1;
cin>>s1;

within the class
cin.operator>>(s1);

operator>>(cin,s1);
```

operator>>(istream &in,Student &s1)

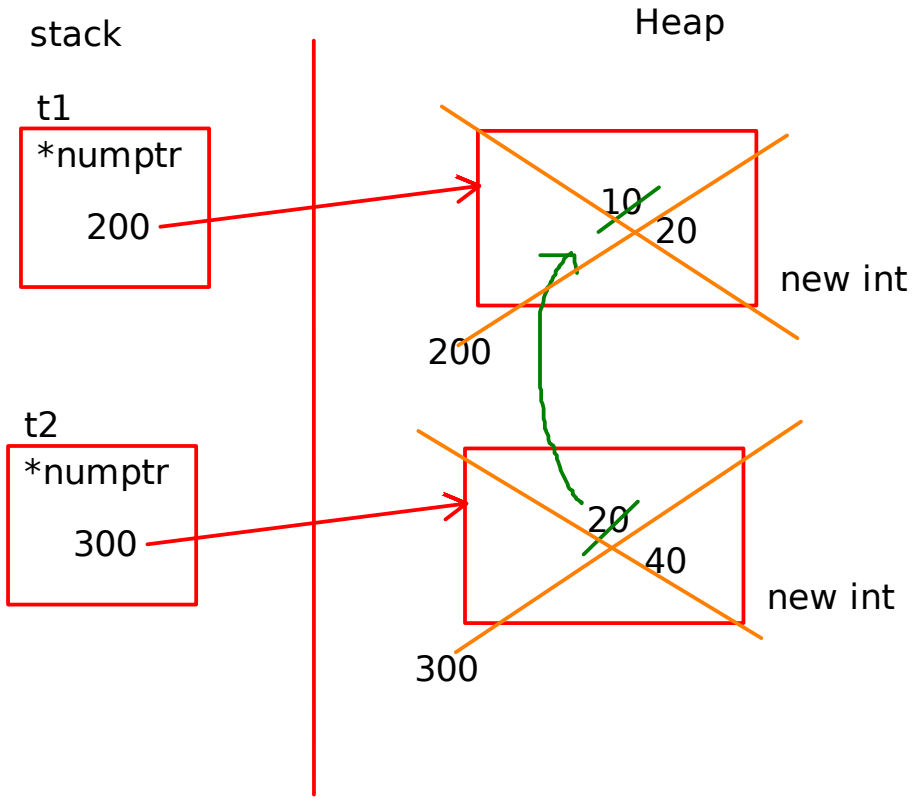



```
Test t1; // 10
Test t2(20); // 20
t1=t2; // copy -> Assignment

cout<<t2<<endl; //20
cout<<t1<<endl; //20
```




```
Test t1; // 10
Test t2(20); // 20

// t1 = t2; // Assignment
```

```
Test t1;

Test t2(20);

Test t3 = t2; // initialization

int num1 = 10; // initialization
num1 = 20; // assignment
```




```
Test t1; // 10
Test t2(20); // 20

t1 = t2; // Assignment
```

Manipulators are used to manipulate the oaniutput
-> Output formatting

Manipulators are of 2 types
1. Without arguments
2. With arguments