```
vector

customer{
vector<Product*> pp;
}
```

```
enum AccountType{
Savings,
Current,
Dmat
}

class Account{
AccountType type;
}
```

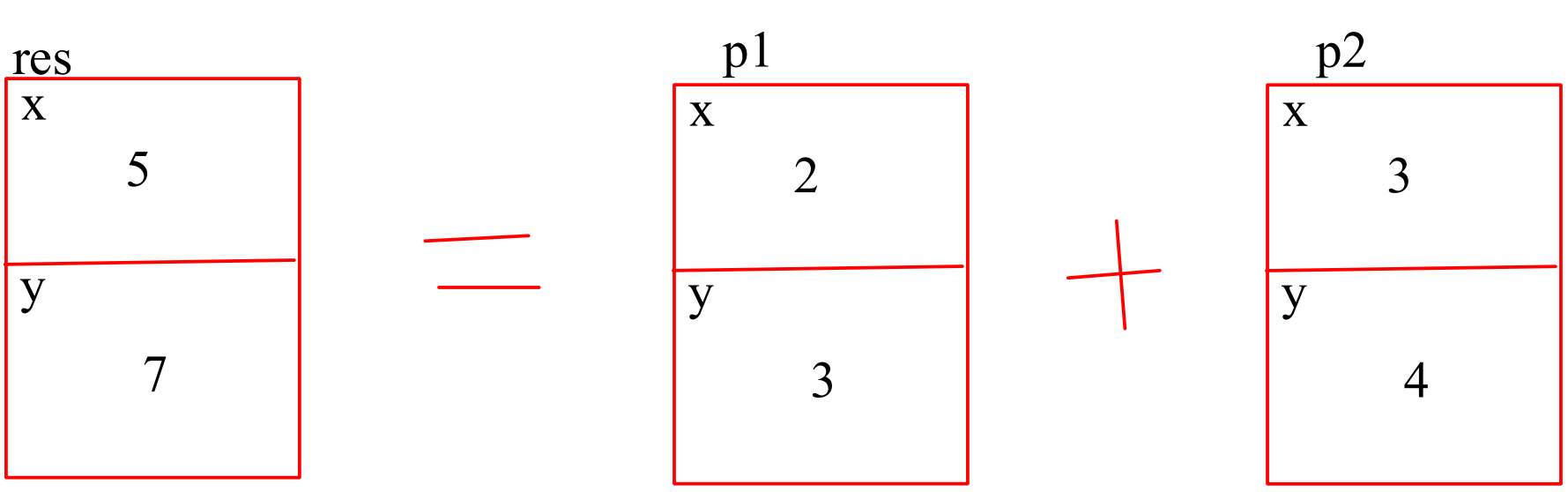
```
class Employee{
id,
name
salary

Employee(){
id=0;
salary = 500;
}
Employee(int id,double salary){
this->id = id;
this->salary = salary;
}
}
```

```
new Employee ()

loadData(){
Employee * e = new Employee();
e->setId(stoi(id));
e->
}
```

Operator Overloading

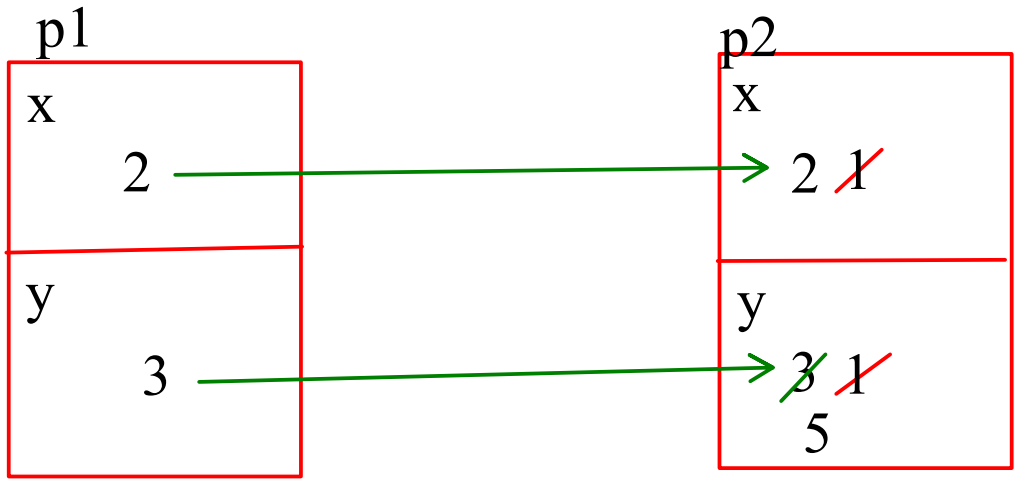
```
10 + 20

p1 + p2

e1 + e2
```

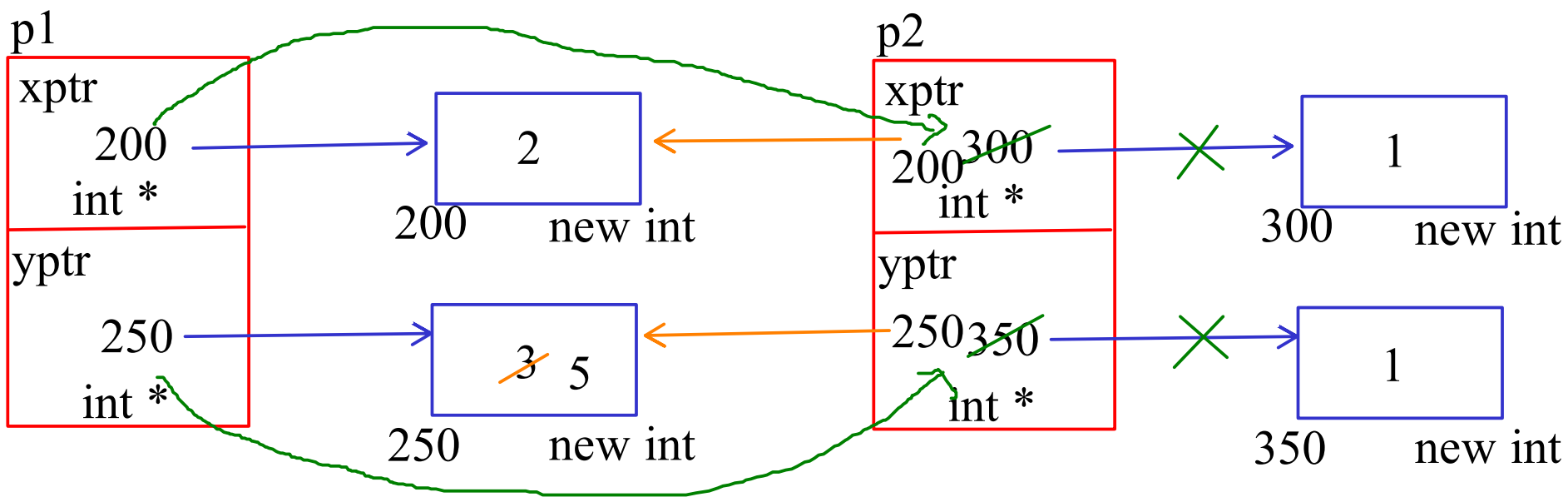
```
Point p1;
Point p2;

Employee e1;
Employee e2;
```

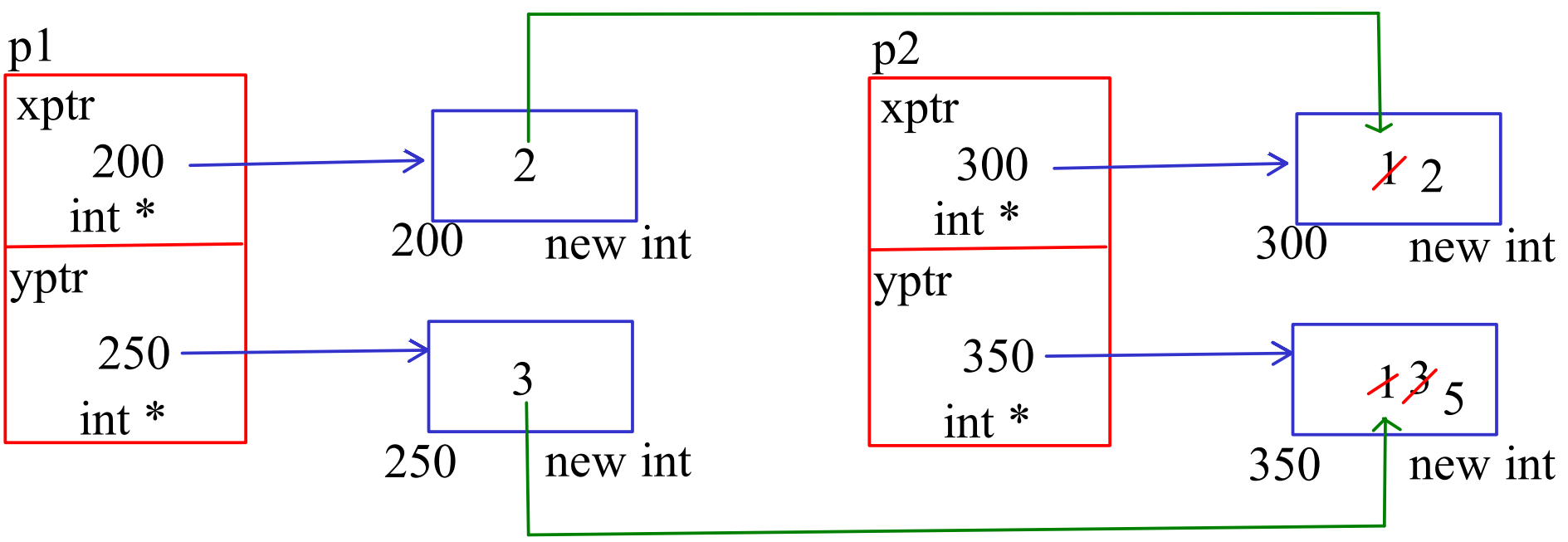


res.x = p1.x + p2.x
res.y = p1.y + p2.y

```
<<
>>
=
()
```







p2 -> this
p1 -> p

*p2.xptr = *p1.xptr
*p2.yptr = *p1.yptr

*xptr = *p.xptr
*yptr = *p.yptr

\+
>>
<<
=
[]
()

Person
Employee, Student

vector<Person *>
Product* arr[3]
book, tape

Customer
<Product>

vector<Person*> personList
vector<Product *> productList

vector<Person*> personList
vector<Product *> productList

Person *p = personList[0]
Product *p = productList[0]

Book, Tape
new Book(), new Tape()

typeid(*peroductList[0]) == typeid(Book)

Customer
Product