stack

int num1;

num1

4 bytes

200

int *ptr =

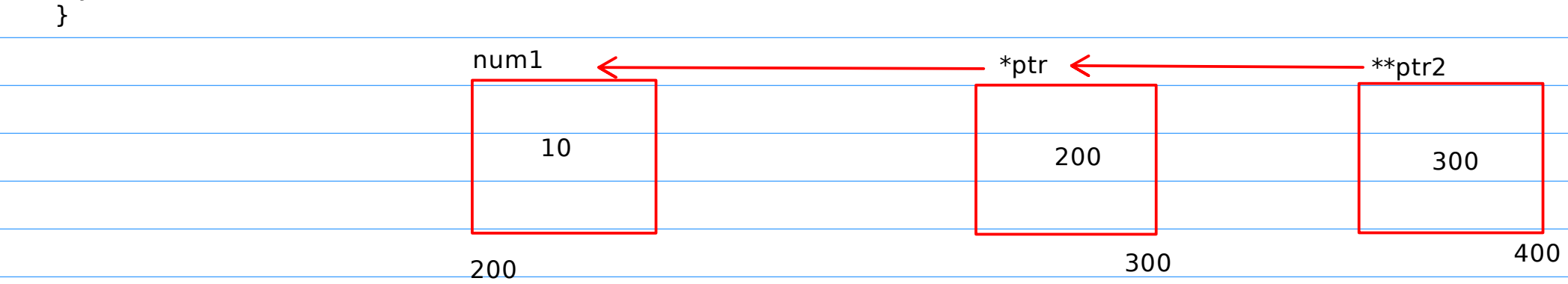
heap

4 bytes

500

```
main(){

int num1 = 10;

myfun(&num1);
}
```

myfun(int *ptr)



num1 -> 10
&num1 -> 200

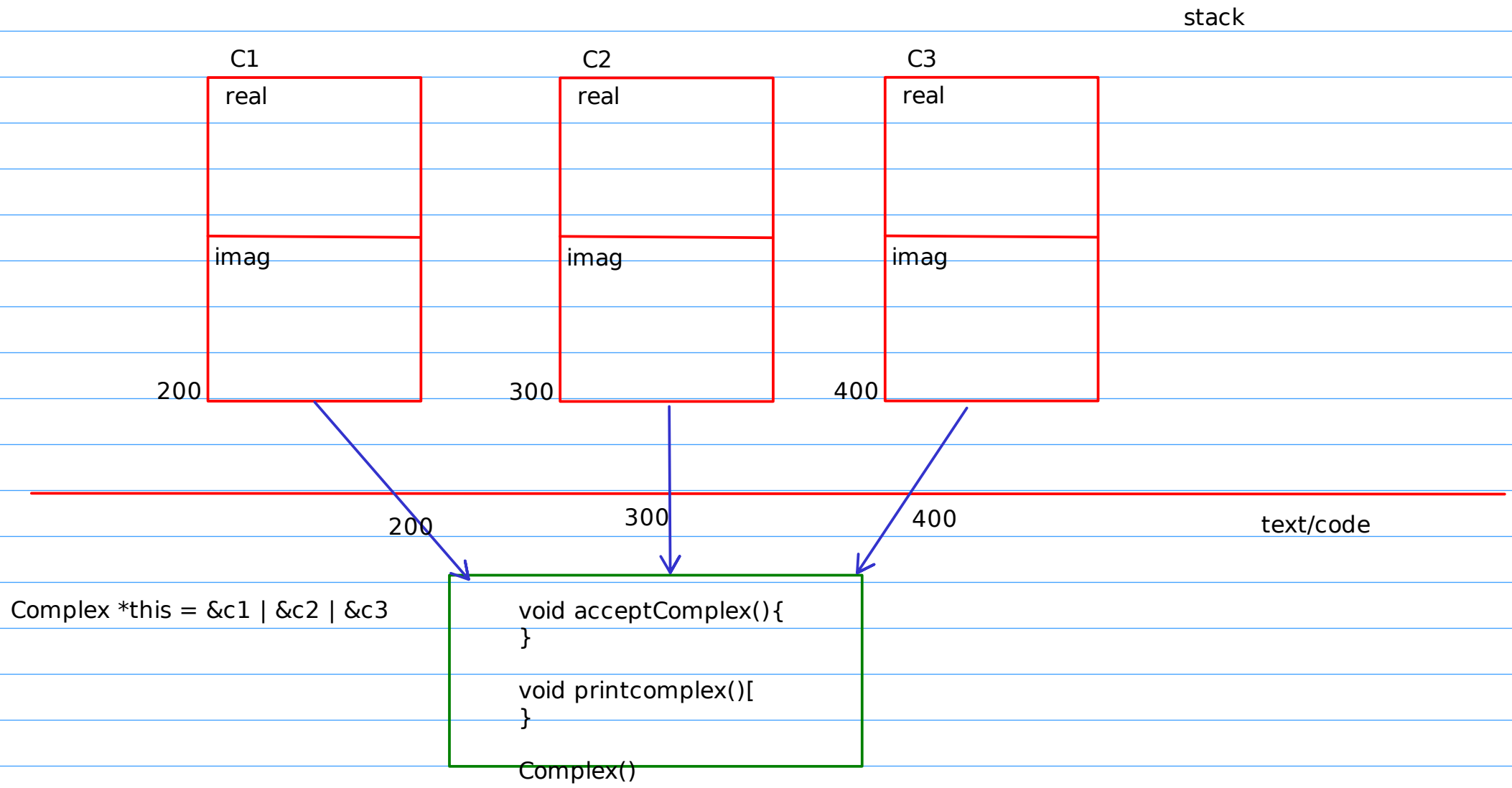
ptr-> 200 <- &num1

&ptr -> 300

*ptr -> 10

int *ptr = &num1; // way1

int *ptr;
ptr = &num1; // way2



Complex *this = &c1 | &c2 | &c3

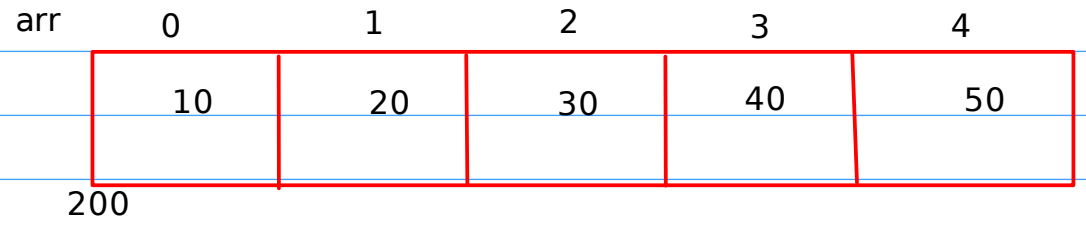
```
void acceptComplex(){
}

void printcomplex()[
}

Complex()
```

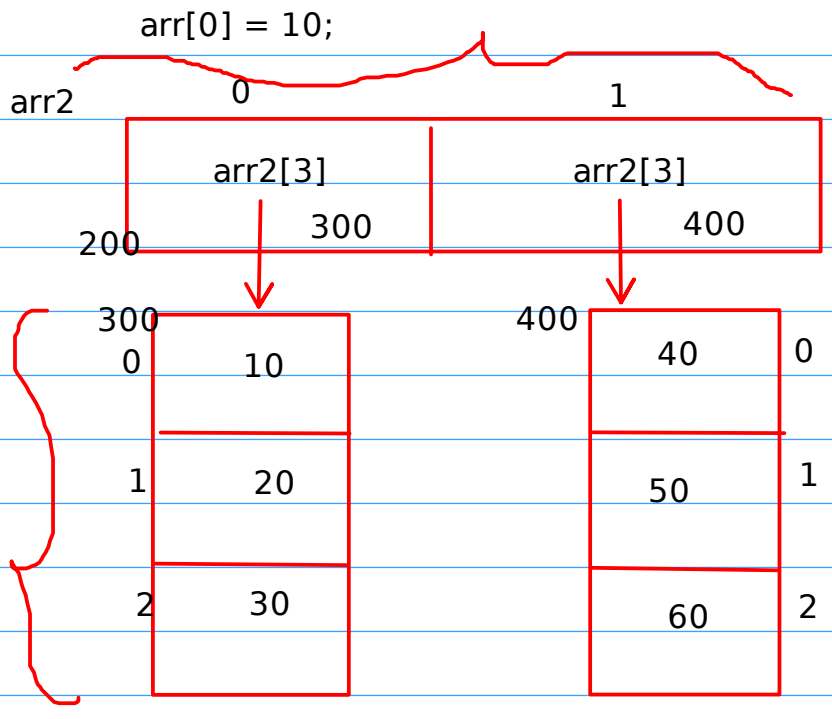
```
class BankAccount{

private:

int accno;
string name;
double balance;

void createAccount(){

}

void dotransactions(){

}
int getAccno(){

}

}
```

```
int main(){

}
```

```
class UPI{

bankAccounts[50]

dopaymemt(){
if(1001 == ba[0].getaccno())

}

}
```

	r	w
accno	Y	N
name	Y	N
balance	Y	Y



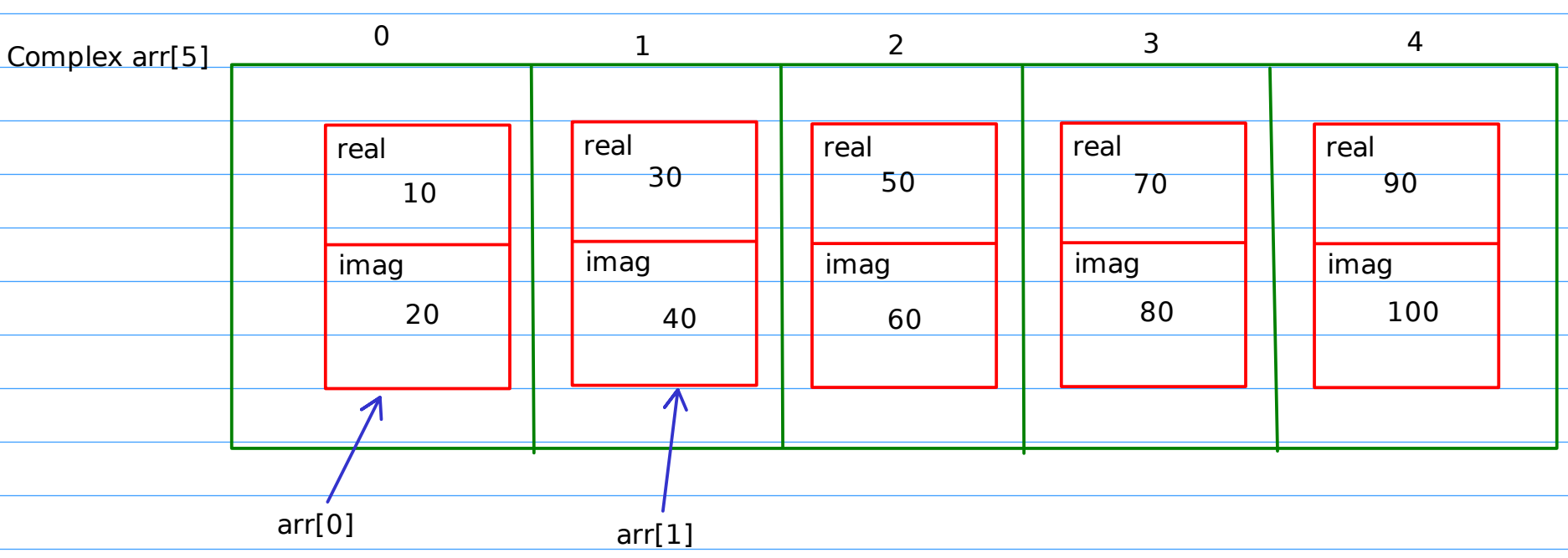


arr2[0]
300

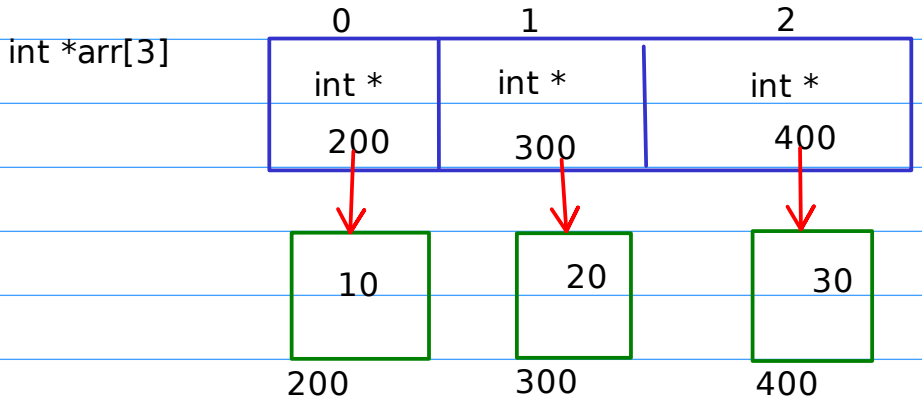
arr2[0][0]
arr2[0][1]
arr2[0][2]

arr2[1]
400

arr2[1][0]
arr2[1][1]
arr2[1][2]



ptr	arr
300[0]	300[0]



int *ptr; can change the address inside ptr as well as the value
only store non constant variable address

const int *ptr; can change the address inside ptr but cannot change the value
can store const and non const variable address

int *const ptr; cannot change the address inside ptr but we can change the value
only store non constant variable address

const int * const ptr we cannot change address inside ptr as well as value
can store const and non const variable address

300[]

element ->
Obj .
*ptr ->