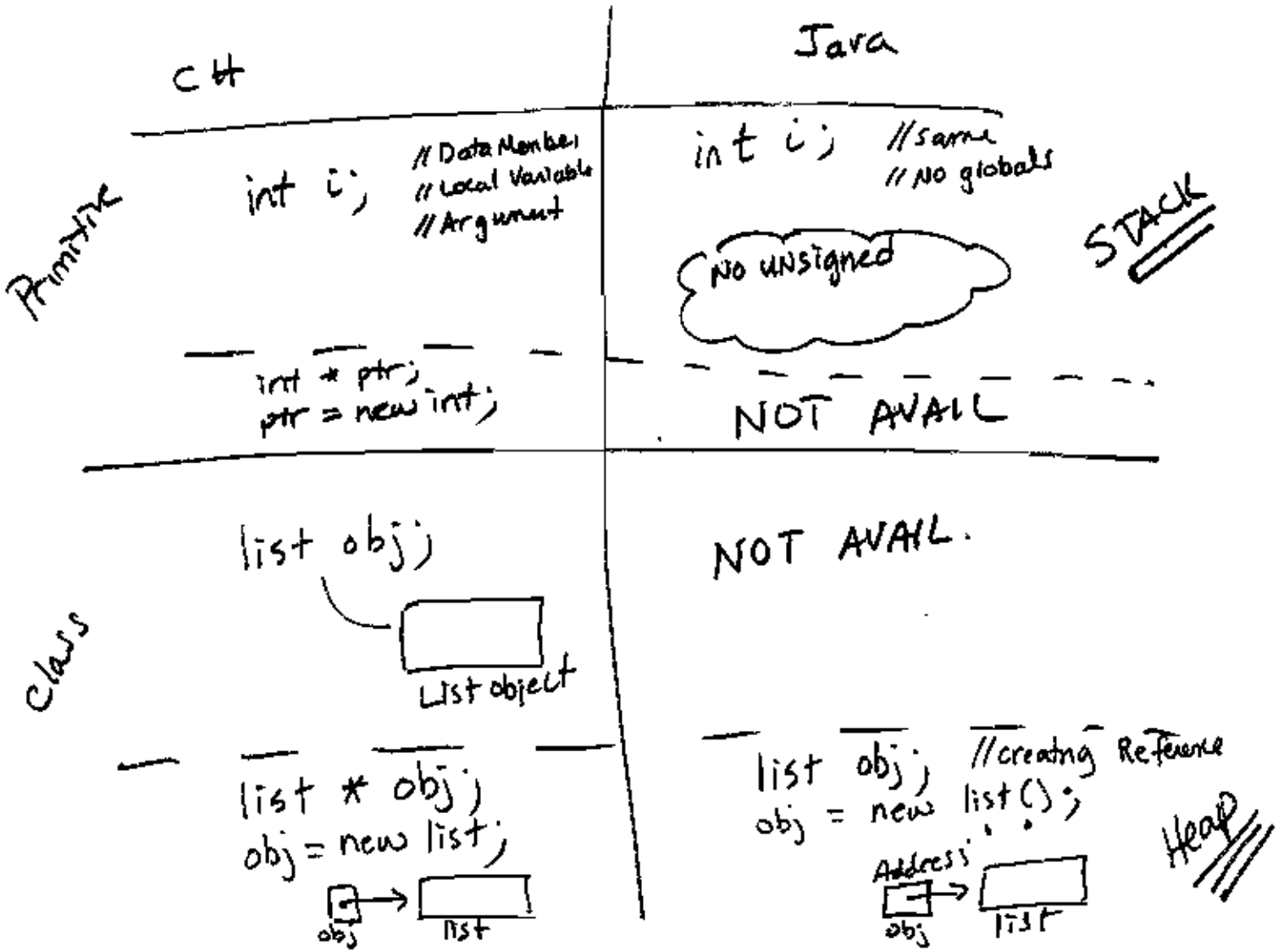

C++
Java
Primitive
int i;
// Data Member
// Local Variable
// Argument
int i;
// same
// No globals
No unsigned
STACK
int * ptr;
ptr = new int;
NOT AVAIL
Class
list obj;
List object
NOT AVAIL.
list * obj;
obj = new list;
obj
list
list obj; //creating Reference
obj = new list();
Address
obj
list
Heap

C++

```
class list
{   public:
        ⋮
    private:
        ⋮
};
```

class interface .h

```
list :: list()
{   ⋮
}
```

class implementation .cpp

Java

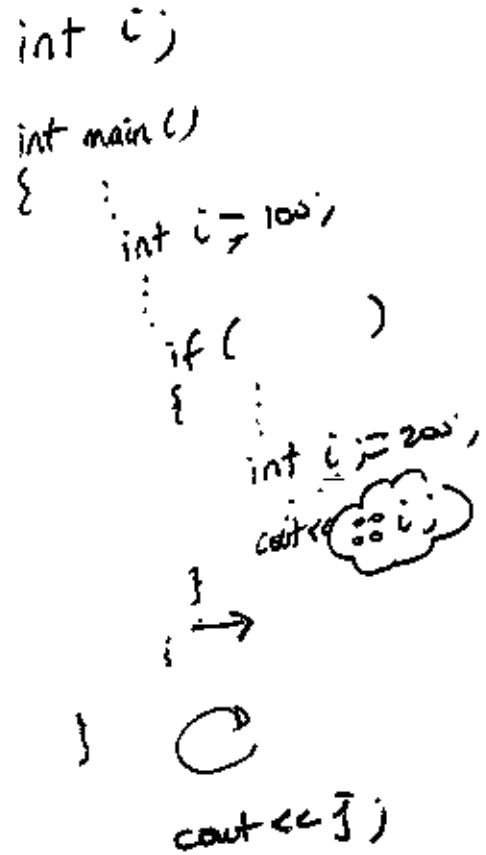
```
class list
{   private int size;
    public list()
    {   ⋮ body
    }
```

field

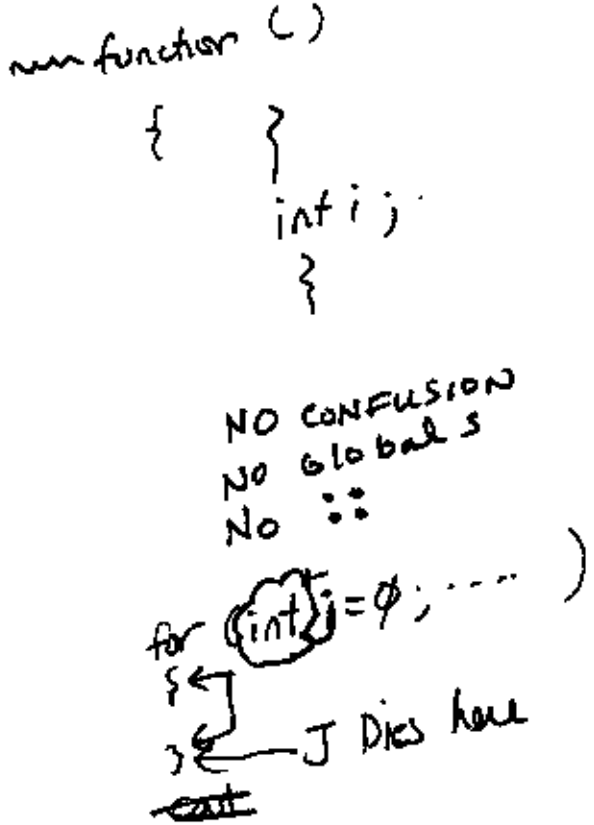
← Method

} NO Semicolon

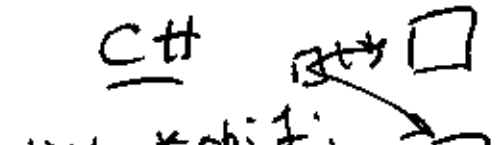
C++


int i;
int main()
{
int i = 100;
if ( )
{
int i = 200;
cout << ::i;
}
}
cout << i;

Java


anyfunction ()
{ }
int i;
NO CONFUSION
NO globals
NO ::
for (int i = 0; ...)
{
}
J Dies here
cout

C++



```
list *obj1;
list *obj2;
obj1 = new list;
obj2 = new list;
```

```
obj1 = obj2;
```

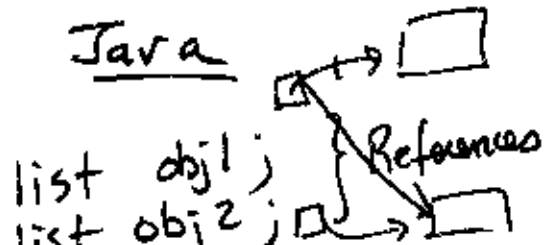
What is this?

Pointer Arith.

if (obj1 == obj2)

ARE WE POINTING TO The Same Place?

Java



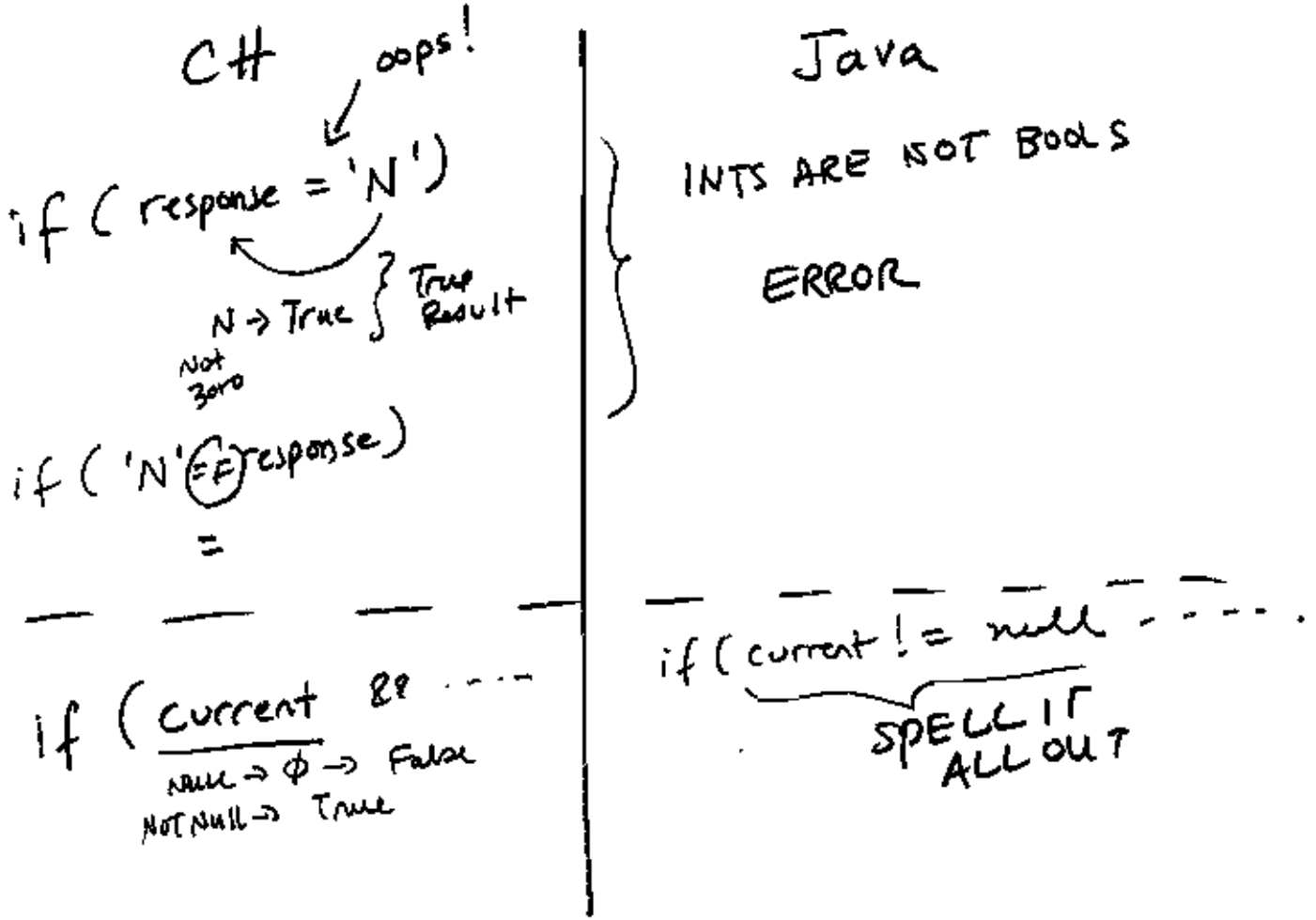

```
list obj1;
list obj2;
obj1 = new list();
obj2 = new list();
```

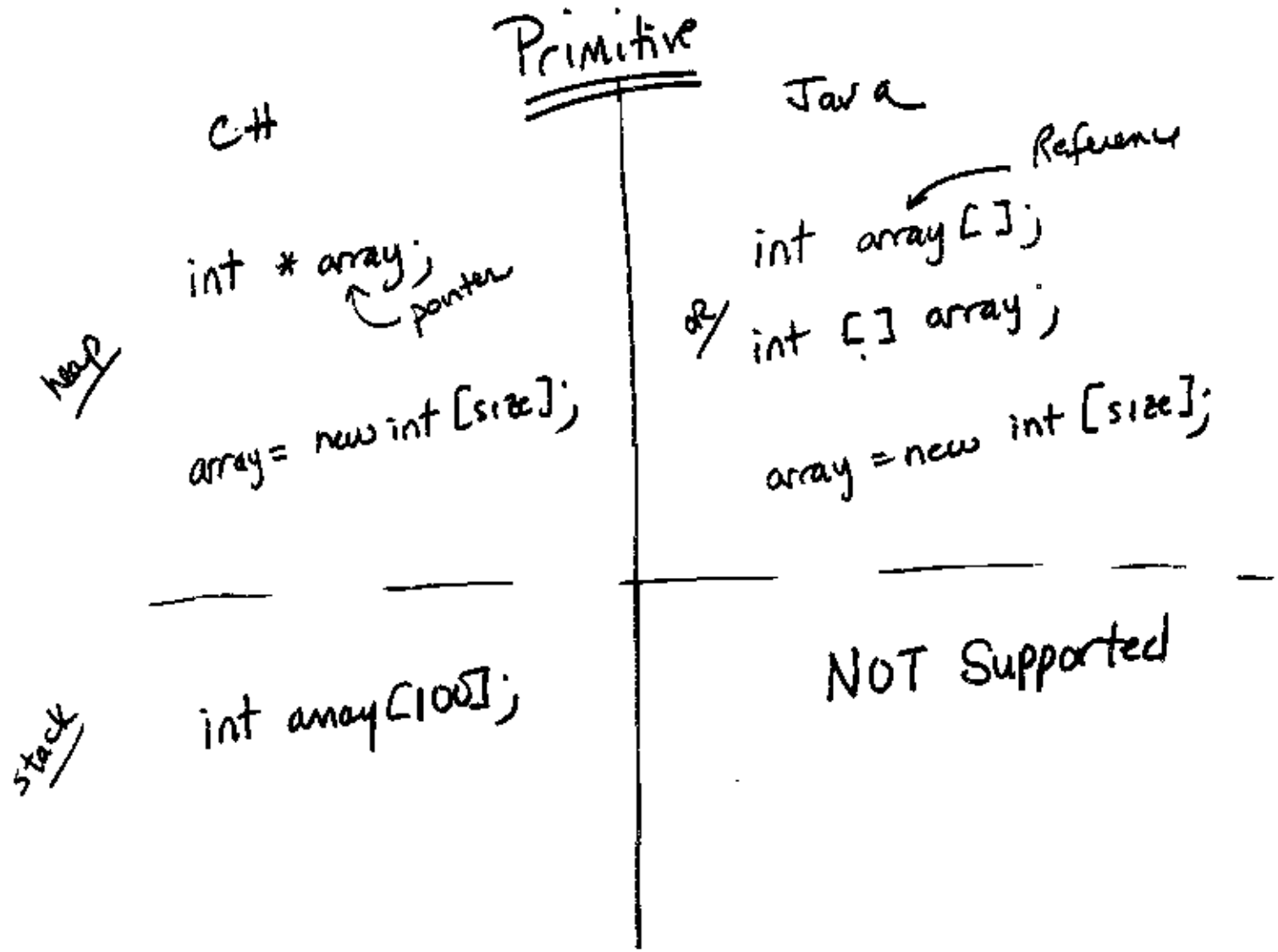
```
obj1 = obj2;
```

Copies an Address

if (obj1 == obj2)

ARE WE Refering to the same place


C++
oops!
if (response = 'N')
N → True
True Result
Not Zero
if ('N' == response)
=
if (current !=
NULL → 0 → False
NOT NULL → True
Java
INTS ARE NOT BOOLS
ERROR
if (current != null ....
SPELL IT ALL OUT


Primitive
C++
Java
Reference
int * array;
pointer
int array[];
or
int [] array;
heap
array = new int [size];
array = new int [size];
stack
int anay[100];
NOT Supported

Class Types

C++

```
list ** array;
array = new list*[size];
for (int i=0; i<size; i++)
    array[i] = new list;
```


Java

```
list array[];
array = new list[size];
for (int i=0; i<size; i++)
    array[i] = new list();
```

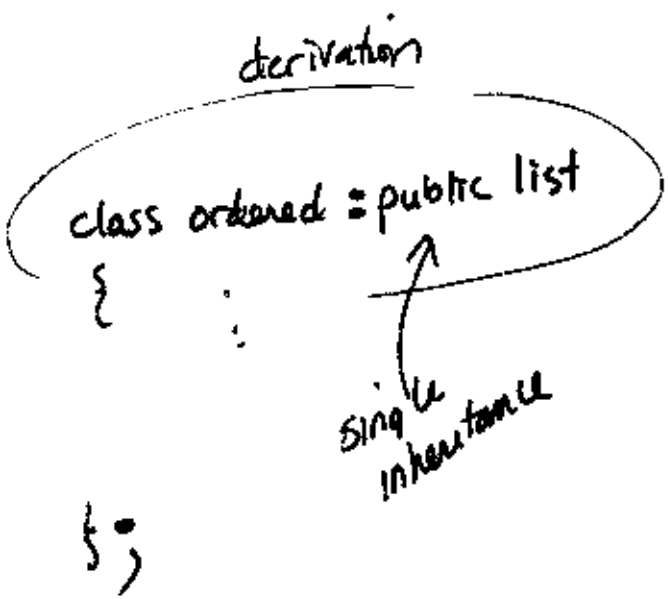

A reference to a list

or

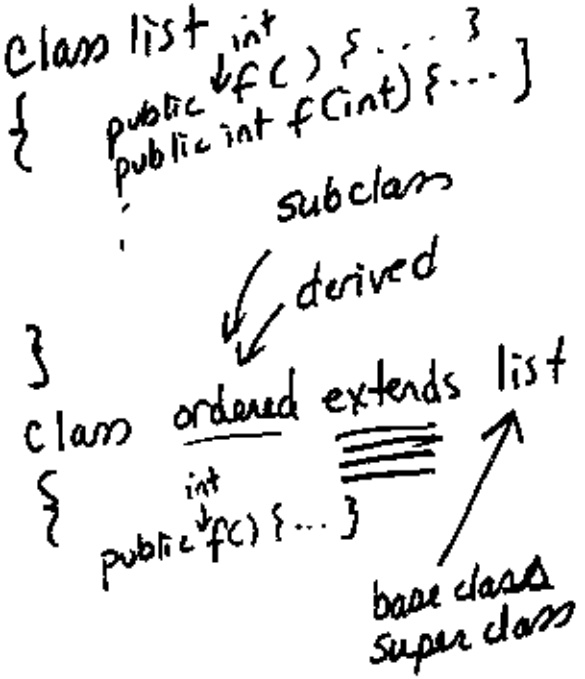
```
list [] array;
```

Pseudocode - Java Not Supported

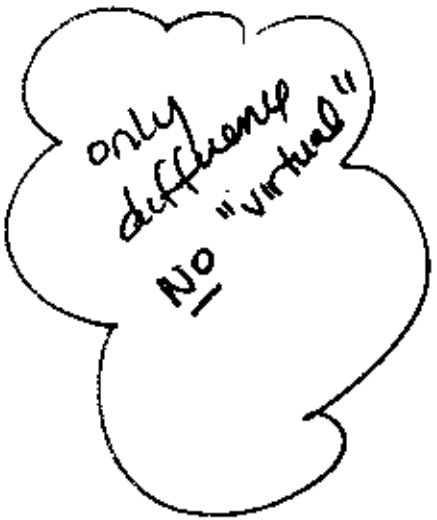
C++


derivation
class ordered : public list
{
:
};
single inheritance

Java

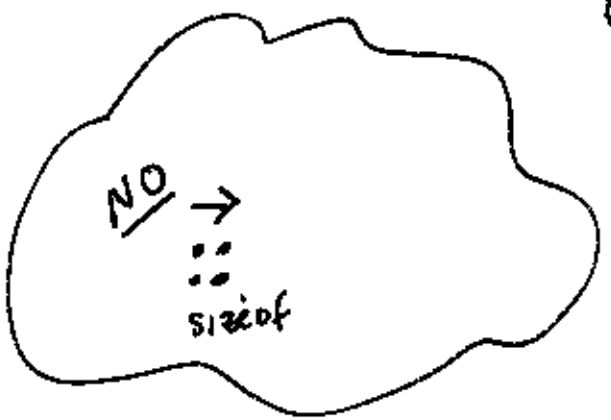

class list
{ public int f() {...}
public int f(int) {...}
:
}
subclass
derived
class ordered extends list
{
public int f() {...}
}
base class
super class
single inheritance

Java




```
list ref;
ref = new ordered();
ref.f(); // call
```

Dynamic Bind's



Recursion

C++

```
void tree::
    copy (node *& newroot,
          node * source)
{
    if (source)
    {
        newroot = new node;
        // copy data
        copy (newroot->left,
              source->left);
        copy (newroot->right,
              source->right);
    }
    else newroot = NULL;
}
```

Public

```
^node copy (node source)
{   node newroot;
    if (source != null)
    {
        newroot = new node();
        newroot->left = copy (source->left);
        newroot->right = copy (....
    }
    return newroot;
}
```