

Today!

• Delay of HW #1 to 10/20

- Structures
- Arrays of structures

Structures

- Group different types of data
- C & C++
- Simplify
 - Grouping of Arrays
 - Arrays of data (Pass by reference)
 - Passing of data (Pass by reference)

← A new data type name

Struct Inventory

should be constants

char name [21];

char description [131];

int quantity; [13];

char barcode [13];

float price;

Members

10/10/2020

*include

```

const int SIZE_NAME = 21;
const int SIZE_DESC = 131;
const int SIZE_BAR = 13;

```

struct Inventory

```

{
float price;
int quantity;
char name [SIZE_NAME];
char description [SIZE_DESC];
char barcode [SIZE_BAR];
}

```

Better

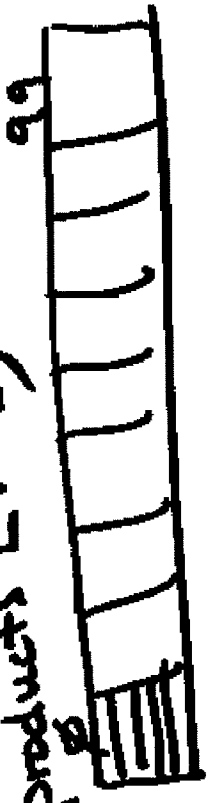
BEFORE
MAIN

```

}
int main()
{
Inventory product; // data type variable
Inventory all_products [100];
int num_prod = 0;
}

```

price
quantity
name
desc.
barcode



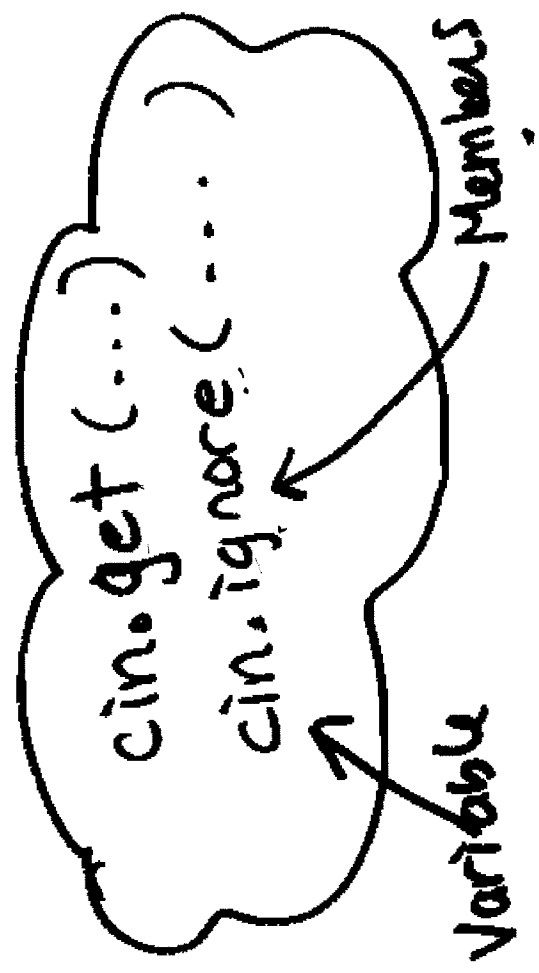


```
Inventory product;
cout << "Please enter price ";
cin >> product.price;
cin.ignore();
```

Direct Member Access Operator

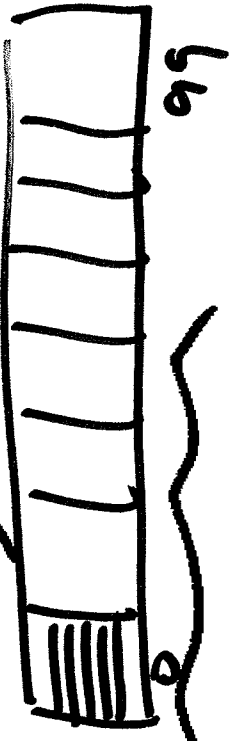
istream cin;

Variable • Member



```
cout << " please enter product name: ";  
// cin >> product.name; // 1 word name only  
cin.get ( product.name, SIZE_NAME);  
cin.ignore (100, '\n');
```

```
Inventory all-products [100];
int num-prod = 0;
```



```
for (int i = 0; i < num-prod; ++i)
```

```
{
  cout << all-products[i].name
```

```
<< '\t';
  << all-products[i].price;
```

```
  cout << endl;
```

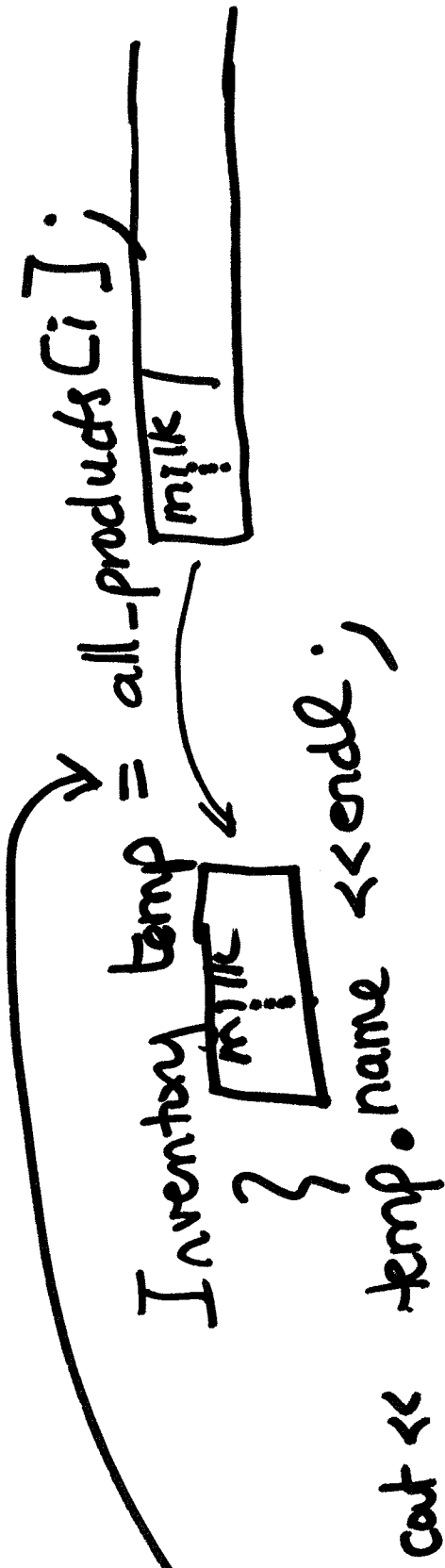
```
}
for (int i = 0; i < num-prod; ++i)
```

```
cout << i;
```



Member
 •
 Variable of type
 Inventory

Would it be easier if...



Yes but... only this week?!

Member wise Copy

Passing

struct

```
void read_inventory (Inventory & item);
```



Pass by Reference

```
Inventory product;
read_inventory (-product);
```

main

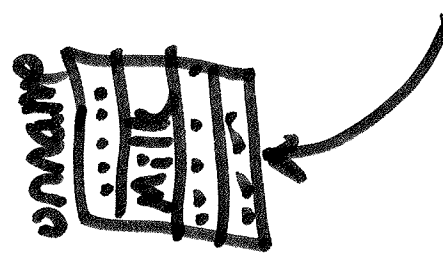
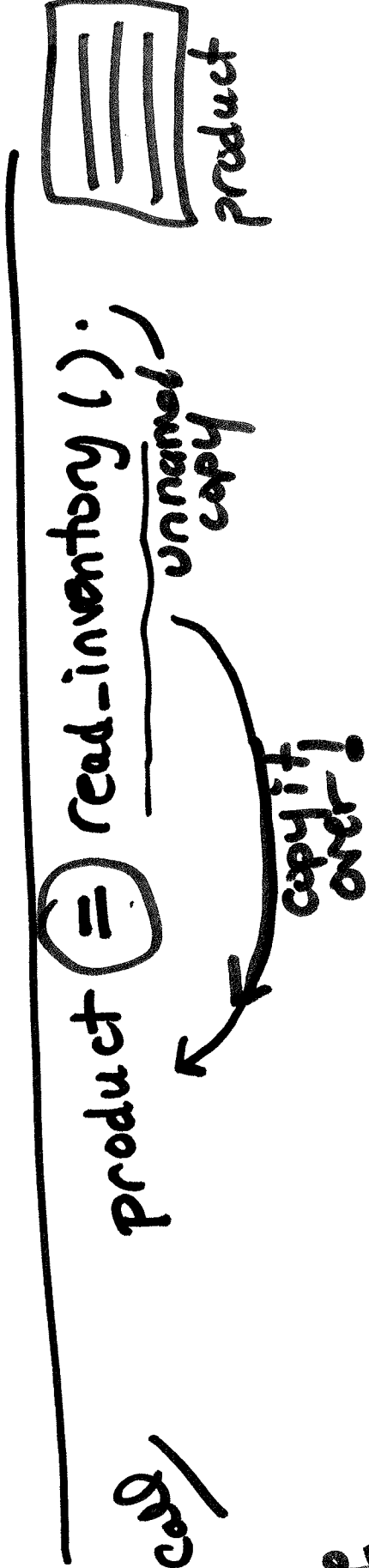
call

```
void read_inventory (Inventory & item)
```

```
{
  cout << "Please enter product name ";
  cin.get (item.name, SIZE_NAME, '\n');
  cin.ignore (100, '\n');
  cout << "Please enter the price ";
  cin >> item.price;
  cin.ignore();
}
```

}

Pass by Value - "what if"
 Return
 Inventory read-inventory();



```

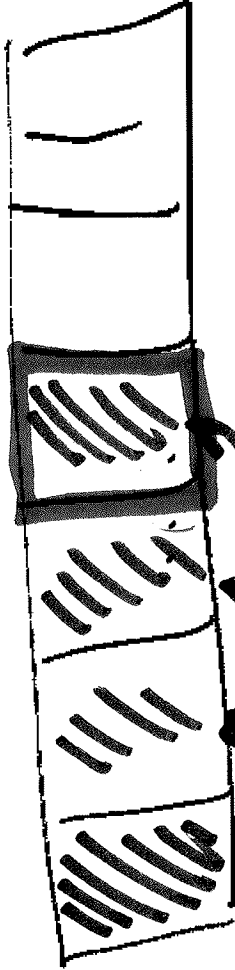
Inventory read-inventory ()
{
  Inventory item;
  cout << "Please enter product name";
  cin.get (item.name, SIZE_NAME);
  cin.ignore (100, '\n');
  return item;
}
  
```

6

read-inventory (all-products [numprod])
 an inventory item
 + numprod;



main

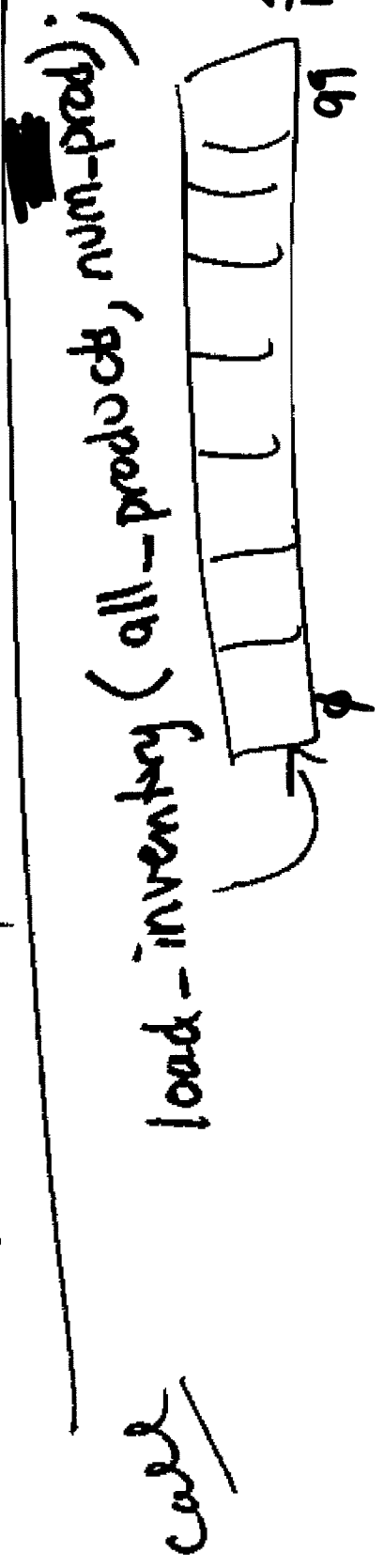


1 numprod

```

void load_inventory (Inventory array[],
                    int & num products)
{
    char response;
    if (num >= 100) return;
    do
    {
        read_inventory (array[num]);
        ++num;
        cout << "Is there another product?" << endl;
        cin >> response;
        if (cin.get() != '\n')
            continue;
        while (response != 'n');
    } while (num < 100);
}

```



"private" 8

"class"

Class

data types

- Grouping of data types
- Create Operations that work on the data

• OOP

"public"

"struct"

Structs

- IN C Language
- Just Group data

100% Same

