Object Oriented Programming

What is an object ??

↳ an real-world entity

Class : collection of a such entity.

Q. Calculate the avg. height of the students ??

→ collection of heights

array ⟷ float

Data Types / Instruction

- bool
- int
- float / double
- char / string
- arrays

attributes

Car

- model
- name
- price
- color
- mileage

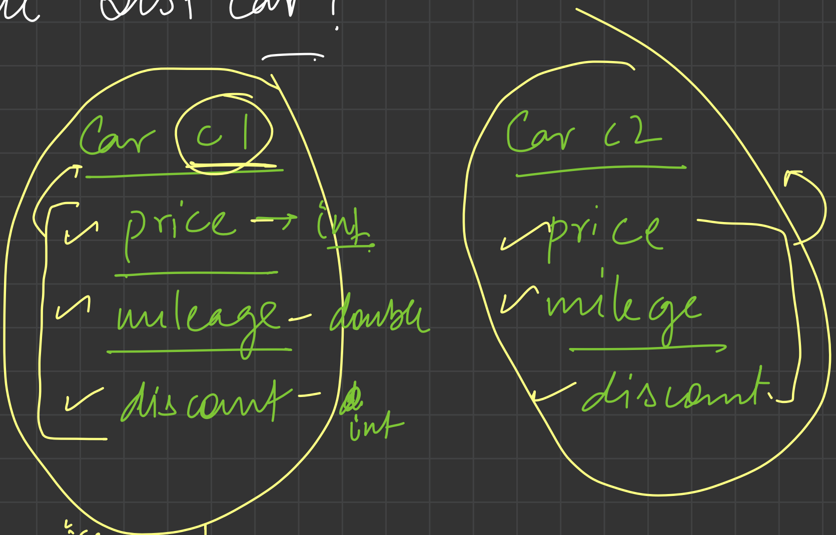
Student

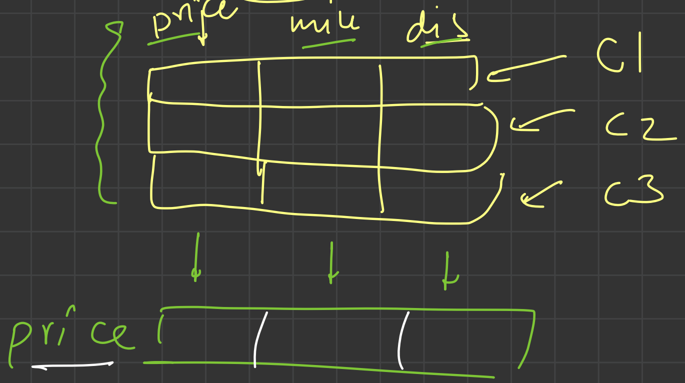
- name
- roll no
- marks
- height
- batch

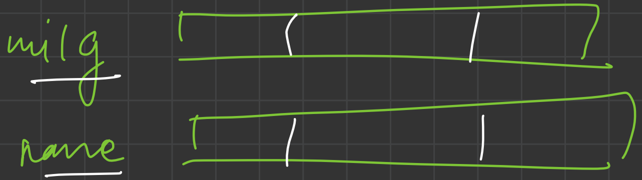
Q. We have to find the best car?

Car

- ~~model~~ : text/string
- ~~name~~ /string
- price ↓
- ~~color~~ / bool
- mileage ↑
- discount ↑







- Structural / Procedural Paradigm
- Object-Oriented
- Functional Paradigm

Objects : - are some real-world entity

- some complex combination of simple features/attributes and functions

Class : blue-print / framework for a type of object

Cars → many different possible

- name
- price
- color
- mileage

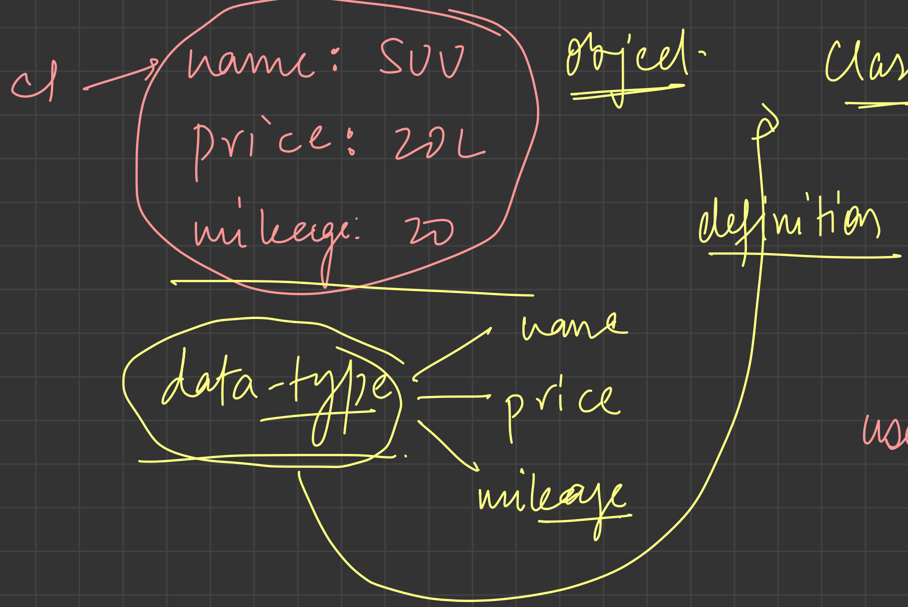
↳ each is unique but has same types of attribu

Class (of cars):

- name
- price
- color
- mileage

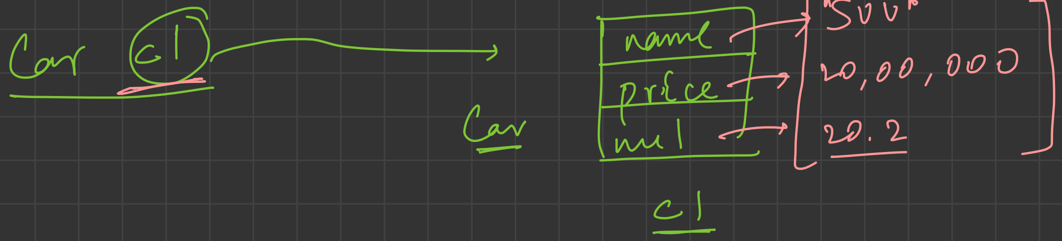
} definition/blue-print of objects

Objects are an instance of a class



```
Class Car { string name;
            int price;
            double mileage;
}
```

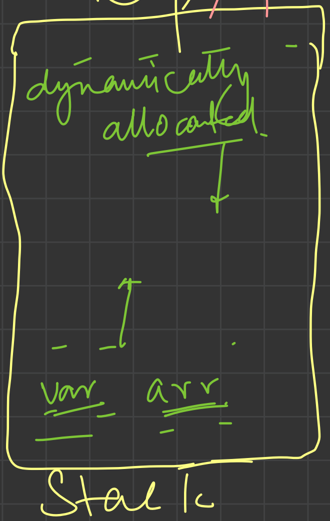
user-defined data type



Dynamic Memory Allocation

- Static/compile Time — variables, arrays — they are assigned memory during compibation i.e. before they are executed
- Dynamic/Run-Time → these memory are allocated when needed/asked for

Heap/free-store



4 bytes

int n; → static

int arr[m]; → static

m×4 bytes

→ get some mem (n-bytes) → dynamic

- we can allocate memory dynamically
- we cannot create variables dynamically

- we need memory to store an int

4 bytes

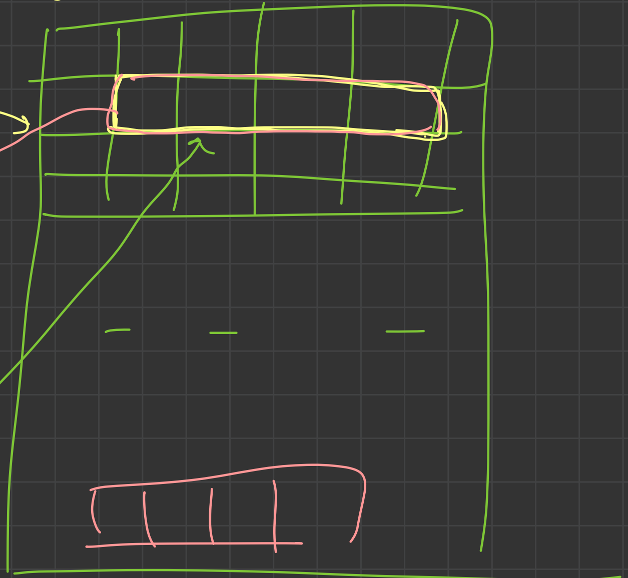
instructions
— get memory for int

int *dint;

dint =

address

*dint



Dynamic allocation requires two-step :

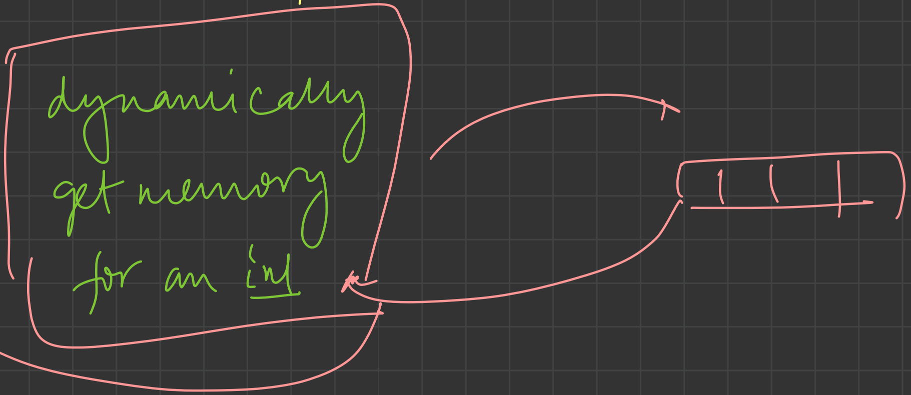
① creating/getting the memory dynamically (address)

② Storing the address in a pointer

```
int *ptr;

ptr = address

*ptr
```


dynamically
get memory
for an int