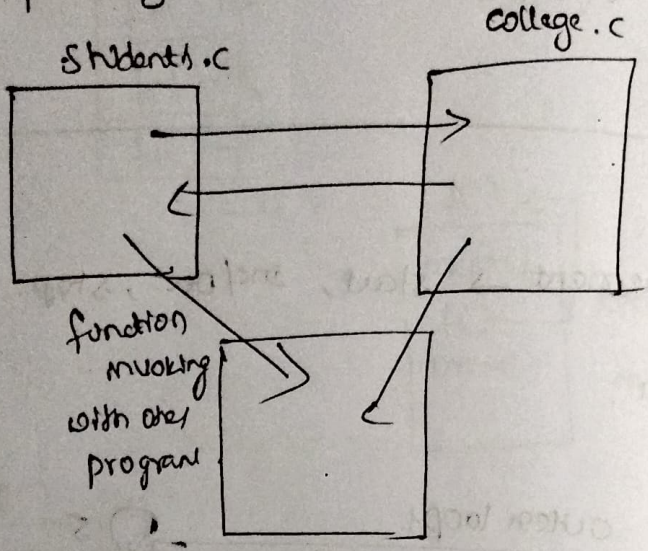
OOPs

10 Sep 2025

Why we need OOPs

Drawback of c language



Ex! Library management System

Entities → Books
→ Students
→ Staff
→ order (check-in checkout)

Entities / Real objects / Actors. } Class

Operations
- log in and logout } Students
- checkout and checkin
- Add books
- Requirements } By staff
- Students registration

} Function / methods.

What is class?

Technique to encapsulate entities.

Ex! Student – class

- Name
- Age
- USN

} Attributes / properties.

Register()
Graduation()
checkout()

} operations – [function / method]

what is object?

entance of class.

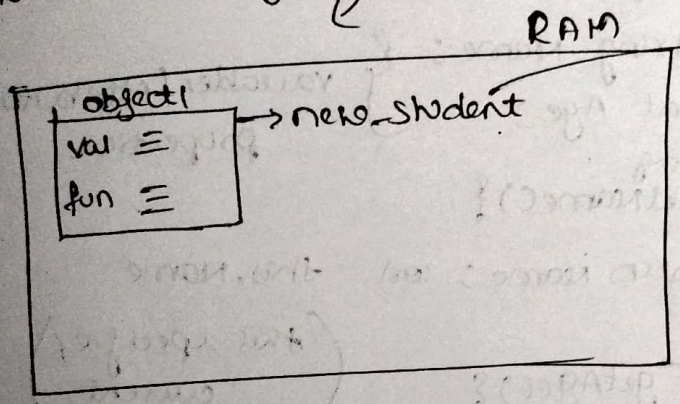
student Student_new = new Student();

↳ class name

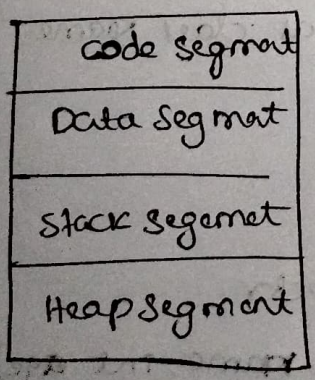
object created in memory/ entance

when we create object

it likes custom datatype.



when code loaded - RAM

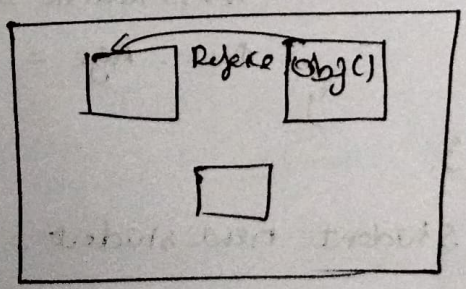


* objects are created by dynamic memory Allocation and created in Heap memory.

* All cleaning process is done by "Garbage collector"

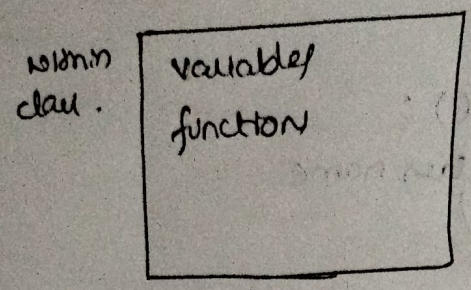
Advantages.

Set of classes - library/ module



Encapsulation → Bundles of both variables and function.

Student class



Constructor:

```
class Student {
    public String Name;
    public int Age;
                                  } variables / Attributes / properties.
    public String getName() {
        return Name; (or) this.Name
    }                    (this specifies current)
    public int getAge() {
        return Age;
    }
    // constructor - same as class Name
    public Student() {
        this.Name = " ";
        this.Age = 0;
    }
    // parametric constructor.
    public Student(String name, int age) {
        this.Name = name;
        this.Age = age;
    }
}
```

Student new_Student = new Student("Vivek", 21);

create a obj in RAM

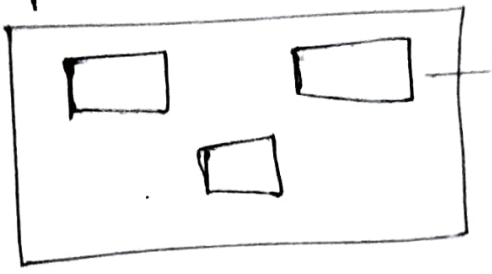
When we compile it converts intermediate - this stored in code segment and objects are stored in Heap.

Access modifiers — public
private
protected
} use for restriction

public — it access Anywhere.

private - it access within class only.

protected - component / module - within module.



Default :— C++ → private.

Key Aspects — [Declaration
Define
} create objects