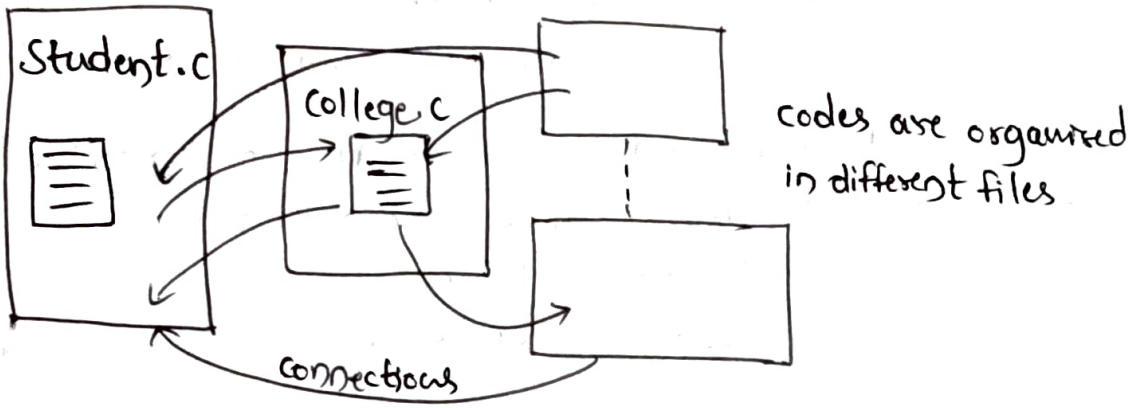
INTRODUTION TO OOP's

*why we need Object Oriented programming
* Real world examples
* key Concepts
→ class, object, encapsulation / Building of data and methods
→ Constructors & Access Modifiers.

Most of the Modern softwares are Built using object Oriented programming

*why we need object oriented programming.

Before using oop's Concept programs of software are written in Different files.



When software data increases or the use of software creates so many connections Between other code files of a software which created complixity to understand the code & Difficulty to modify.

*Maintaince & Management Becomes a Big issue
*To solve this issue object oriented programming is introduced
*Suppose Take any real world activity or concept.
→what are the entities we required to create software for it
- Entities → Real world Object's → Actors

lets take a Library Management System as example

What are the Entities Here?

① Books ③ Staff
② Students ④ Orders

] Analyze the Departments, list all entities.

The task performed by different entities are considered as operations.

Ex: Login / Logout

from Student / Staff point → [checkout of Books, checkIn of Books] → operations or Actions

from Library working staff → [add Books, Request of Books, stocks of Books, New student registration]

All these actions become Functions or methods and All the Entities Become classes.

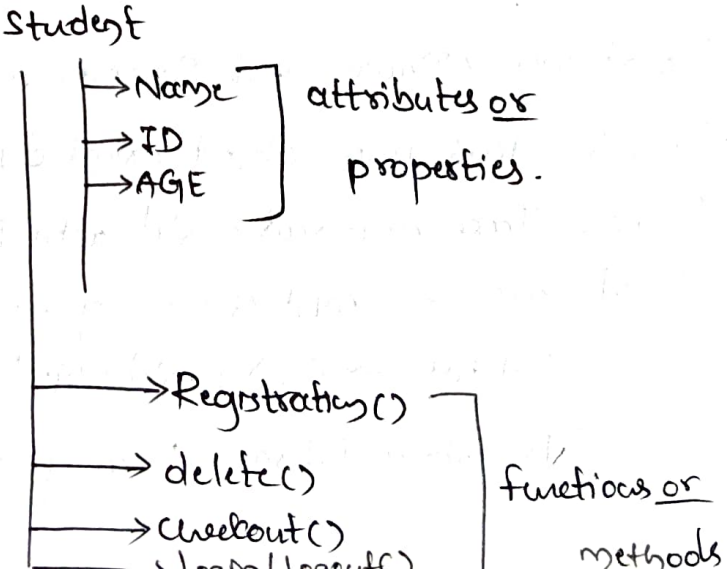
Entities → classes

Entities Actions → Method

What is class?

Encapsulation All the data & Methods related to a particular entity. (Its encapsulating technique)

Example: Student class



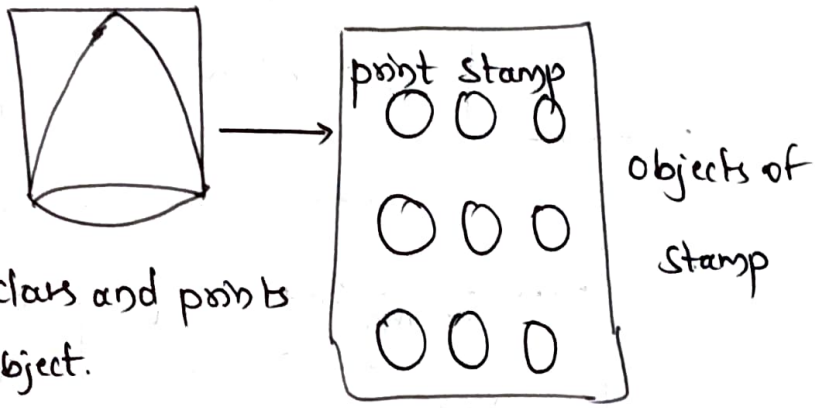
Boundaries are created to create safe softwares.

How to use this class?

Classes are used by creating objects

lets take stamps as example

Stamp Class



objects of Stamp

Here Stamp is class and prints of stamp is called object.

In Memory

keyword used to create object of class

Student Ragu = new Student();

Defines Datatype of Variable or Instance

Variable or Instance

Stored Here

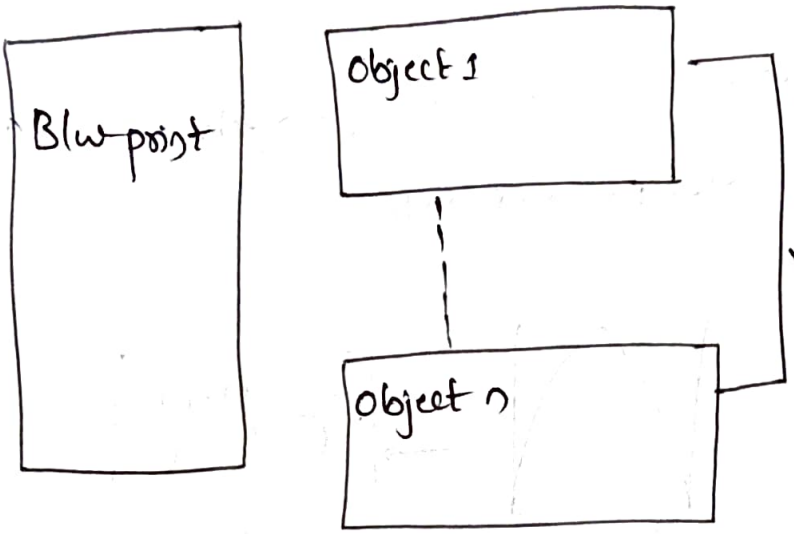
object of class Name student class

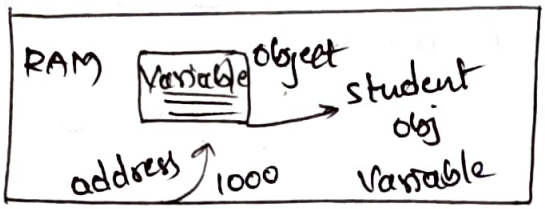
Here Variable 'Ragu' is a variable of type "Student class" which stores the address of object of Student named "Ragu"

we created a New data type

Class is Nothing But Creating Custom Datatype

→ Like this we can create a 'N' number of Instances or objects with the help of class

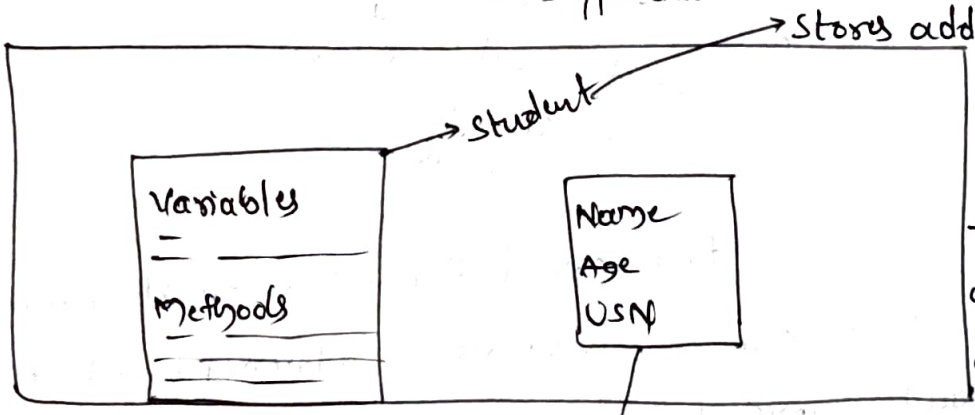




Each Variable stores address of respective objects

Classtype Variable = new ClassName();

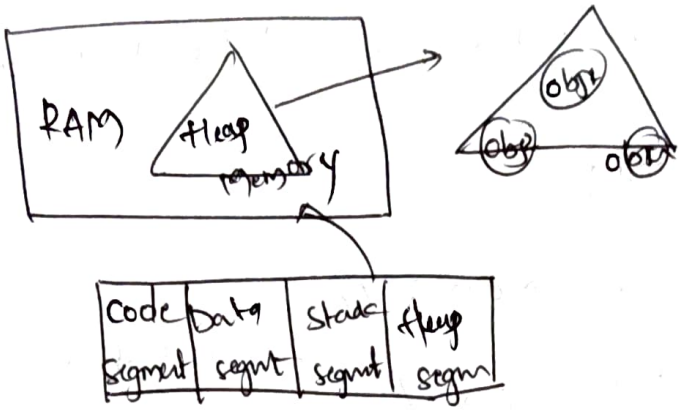
→ Stores the reference address of object of classtype data



Student.name → Accessing

Objects are allocated dynamically (Dynamic Memory Allocation)

So the memory for objects are allocated in RAM in Heap Memory Segement.



Now we learned about classes and objects
we also learned about creating of Instances.

But how objects are deleted (deleting of unused or objects which dont have reference)

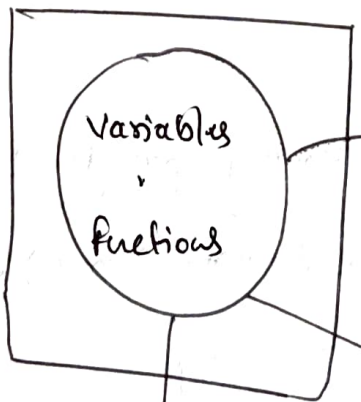
deleting Handled by garbage collectors

Garbage collectors find the unused objects and Cleans from the Memory.

If we have address inside Heap Memory we can Access it

Objects are accessed only if we have address of it

Encapsulation: ⟶ Binding



Binding of methods & variables of class object that can't be accessed without permissions

Together Binding in a memory

Binding of Data & Functions

Coustructors :→ used to set values to dataset of object

```
Class student{

    public string Name;
    public string Age;        // Variables / Attributes / properties

public string getName(){
    return this.Name;         // function
}

    public student(){
        this.Name = " ";
        this.Age = 0;
    }

    or

    public student(string Name, int Age){
        this.Name = Name;
        this.Age = Age;
    }

}
```

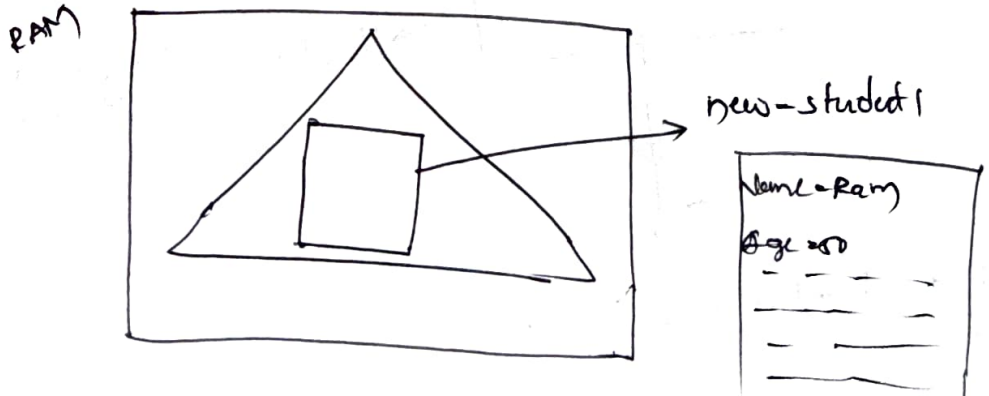
constructors

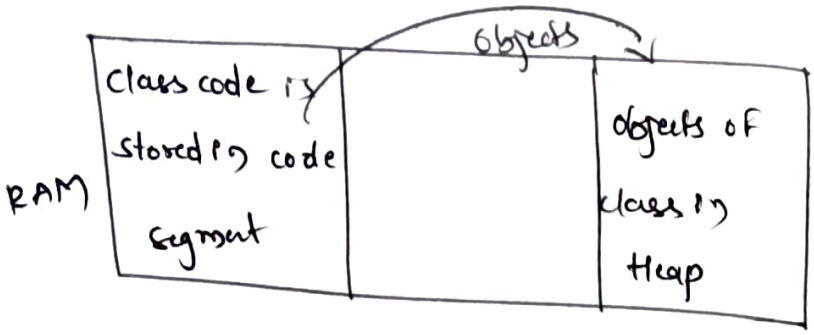
Example for same function name but different behaviors (polymorphism)

Creating obj Reserving space for it

```
Student new_student1 = new student("Ram", 50);
```

Constructors invoked value assigned



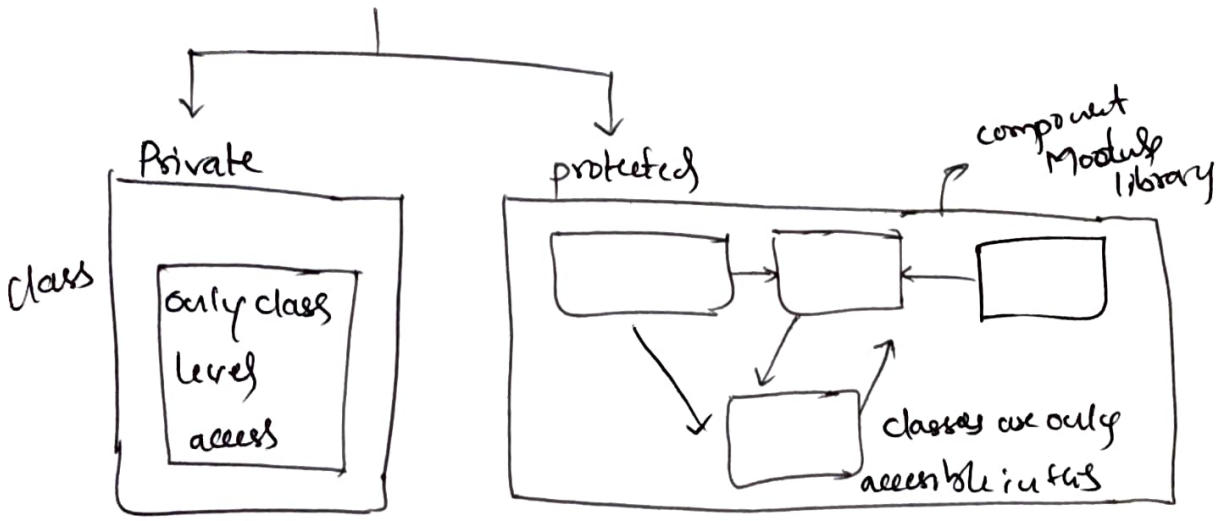


Code always resides in code segment objects in Heap Mapping

Access Modifiers :-

- public → Any one can Access
- private → There are Restrication for Access
- protected → More Restrication for Access

used to protect Data



Interfaces also supported

Access Modifiers → Security Guards