OBJECT ORIENTED PROGRAMMING USING Java

Let's explore technology
together to live in the future

Checkout more on
https://github.com/Sy-hash-collab

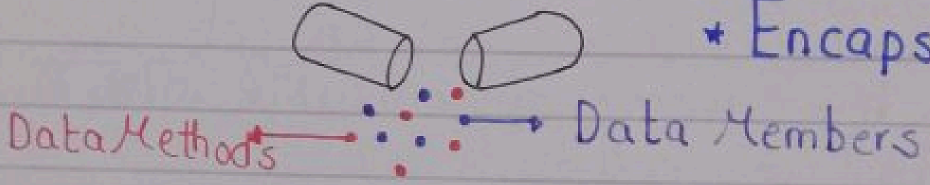
Sy-hash-collab

Encapsulation in Java:

Encapsulation in Java refers to integrating data (variables) and code (methods) into a single unit.

In encapsulation, a class's variables are hidden from other classes and they can only be accessed by methods of the class in which they are found.

"It's an object-oriented procedure of combining the data members (variables) and data methods of the class inside the user-defined class.



* Encapsulation

It's a protective shield that prevents the data from being accessed by the code outside the shield.

=> The meaning of encapsulation is to make sure that "sensitive" data is hidden from users.

It preventsouter class from accessing and changing fields and methods of a class. This also helps to acheive data hiding.

=> To achieve this, you must:

- declare class variables/attributes as private
- provide public get and set methods to "access and update" value of private

We will make a controlled access of it. by using:

- Getter: To access any private member
- Setter: To modify any private member.

Get and Set:

=> As, private variables can only be accessed within the same class. It's possible to access them if we provide public get and set methods.

- get method - returns the variable value
- set method - set the value.

Syntax:

Syntax for both is that they start with either get or set, followed by name of variable with first letter in uppercase.

```
public class Person
{
    private String name;
    // Getter
   public String getName()
    {
       return Name;
    }
  // Setter
  public void setName(String newName)
  {
      this.name = newName;
  }
}
```

```
class Encapsulate
{   // private variables declared
    private String name;
    private int rollno;
    private int age;

  public int getAge() // get method for age
   { return age;}

 public String getName() // get method for name
   { return name;}

 public int getRollno()// get method for rollno
   { return rollno;}

 public void setName(String newName)
     { this.name = newName; }

 public void setAge(int newAge)
     { this.age = newAge; }

 public void setRollno(int newRollno)
     { this.rollno = newRollno; }

public class Main
 { public static void main(String[]args)
     {
          Encapsulate obj = new Encapsulate()
obj.setName("John"); // setting values
obj.setAge(19);          of variables.
obj.setRollno(51)
```

```
 //Displaying values of variables
 System.out.println ("Name:"+ obj.getName(),)
System.out.println("Age:"+ obj.getAge());
System.out.println ("Roll no:"+ obj.getRollno());

  }
}
```

Output:

```
Name : John
Age : 19
Rollno : 51
```

In C, the datamembers are by default public
In C++, the datamembers are bydefault public

In Java, the datamembers are by default "package".

The returntype of getter is same as the returtype of the datamember.

Setter method is always public and void.
It always has one parameter only and the parameter has same datatype as that of datamember.

```
public void setName (parameter)
 {
     this.datamember=Parameter;

  }
```