OOPS2

Agenda

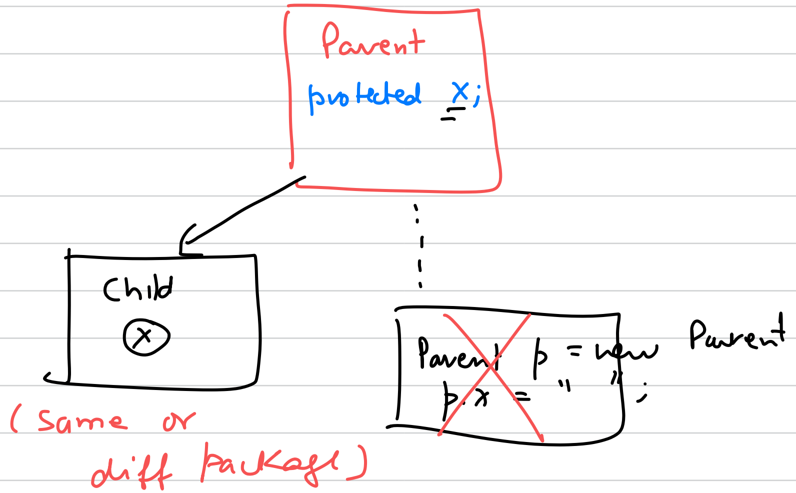
- ✓ Acess Modifiers
- ✓ Getters & Setters
- ✓ Constructors, Overloading
- ✓ Shallow & Deep copy
- Object References, object copy
- Project - GAME

10·25

Acess Modifiers

```
class Player {
    ______   String name
    private  int    guess
             ======
}
```

private	visible only Inside the class
✓ protected	acess in the child class
✓ default	visible within package
public	Visible everywhere

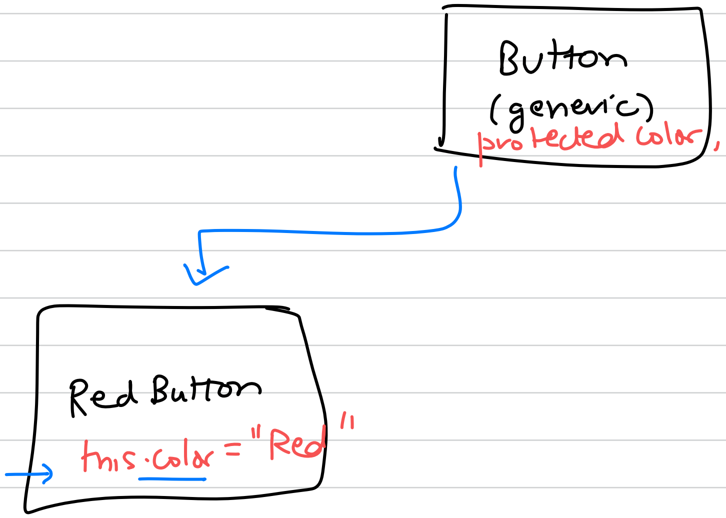


✓ public - The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package

✓ protected - The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

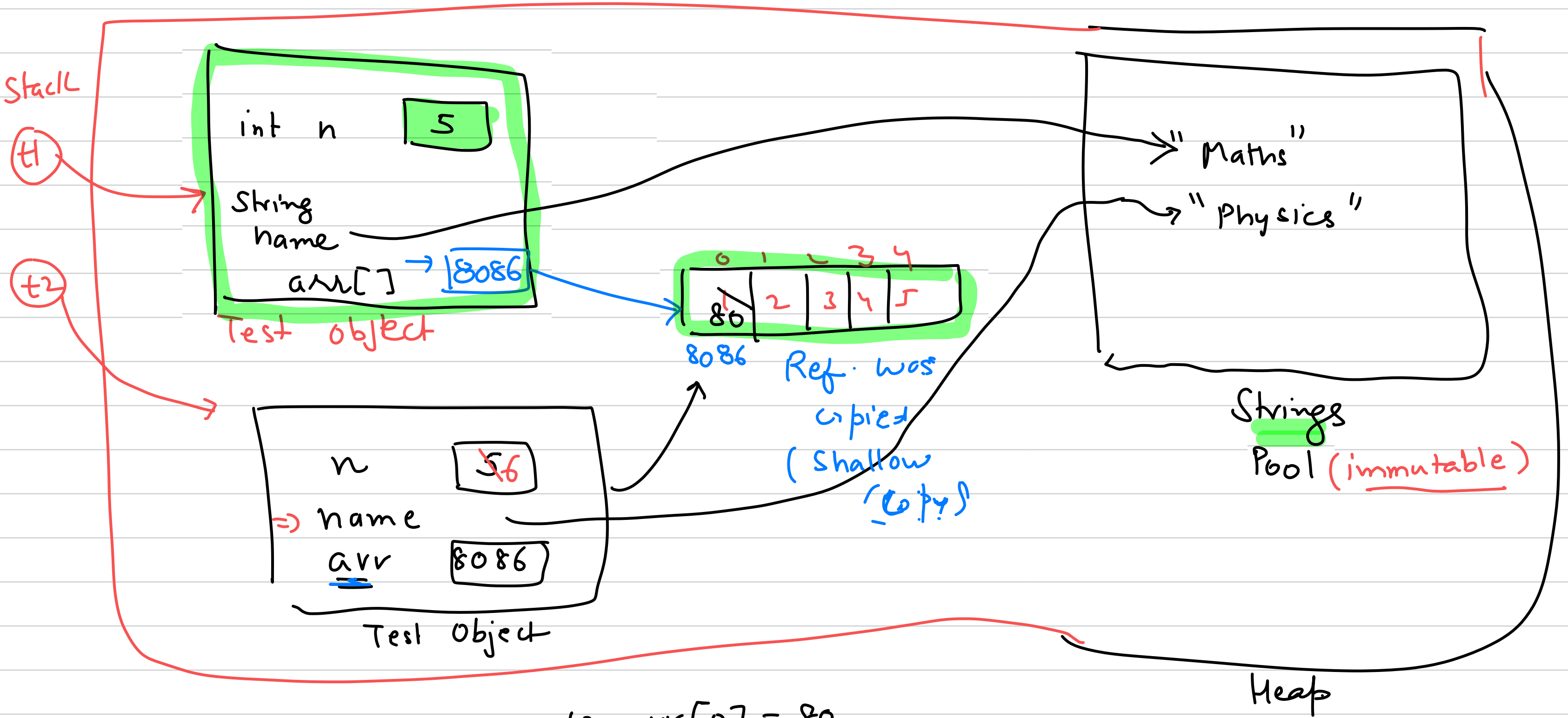
✓ private - The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

✓ default - The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

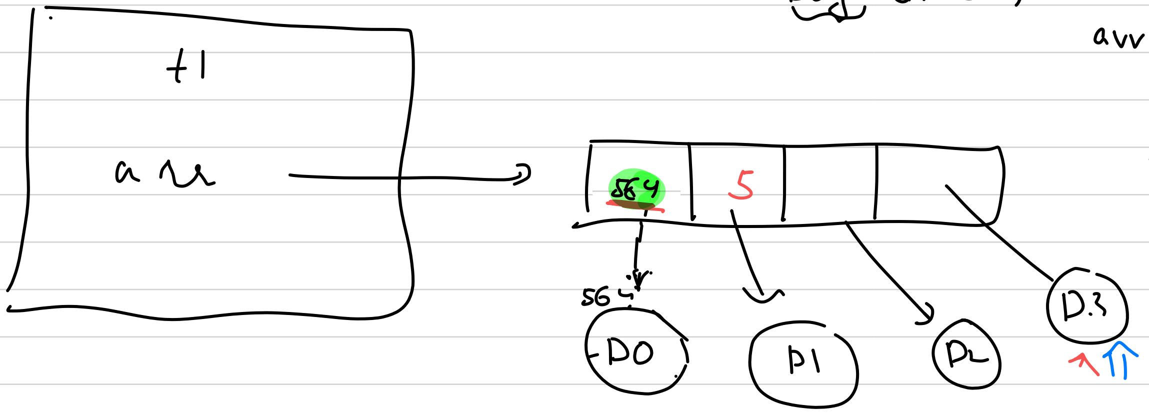


~~button.color = "Red"~~

Shallow Copy vs Deep Copy

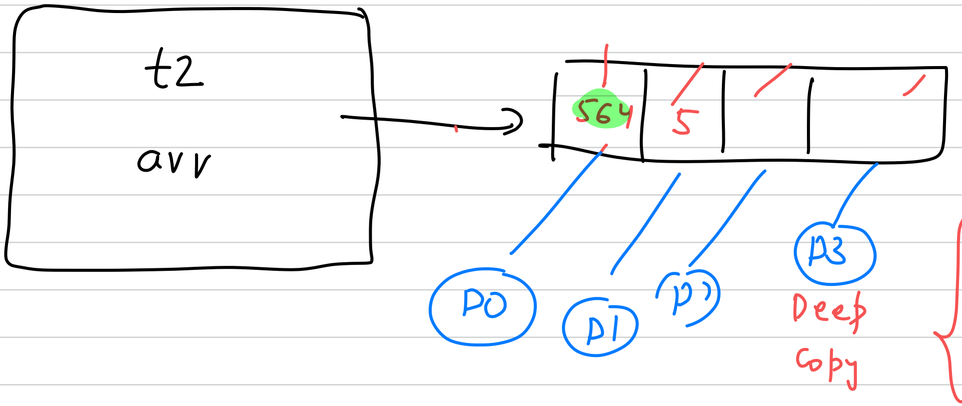


t2. arr[0] = 80



```
Dog arr[];

arr[0] = new Dog();
   ⋮
```

```
Dog {
    dresses: [ | | ]
}
```



```
t2.arr = new Dog[n];
for(i=0; i<n, i++){
    t2.arr[i] = obj.arr[i];
}
```

or ✓ `t2.arr[i] = new Dog(obj.arr[i]);`

Doubt

by Anjum

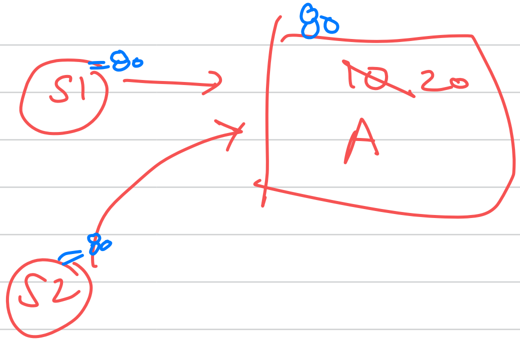
```
Student {
        age
        name
        SayHello()
}
```

Main:

```
Student S1 = new Student (10, "A")
Student S2 = S1;
```

Copying the ref.

$\boxed{S2.age = 20}$



depends how this method is implemented.

Shallow Copy

Student s2 = new Student(s1);

Shallow Copy

A shallow copy of an object is a new object that stores references to the original object's fields. If the field value is a primitive type, the value is copied; however, if the field is an object, then the reference is copied (i.e., the address is copied, not the object itself). This means both the original and the shallow copied object will refer to the same object for their object fields.

Deep Copy

A deep copy of an object copies all of the original's fields, and if the fields are objects, it creates new objects with copies of those fields. A deep copy is fully independent of the original object.

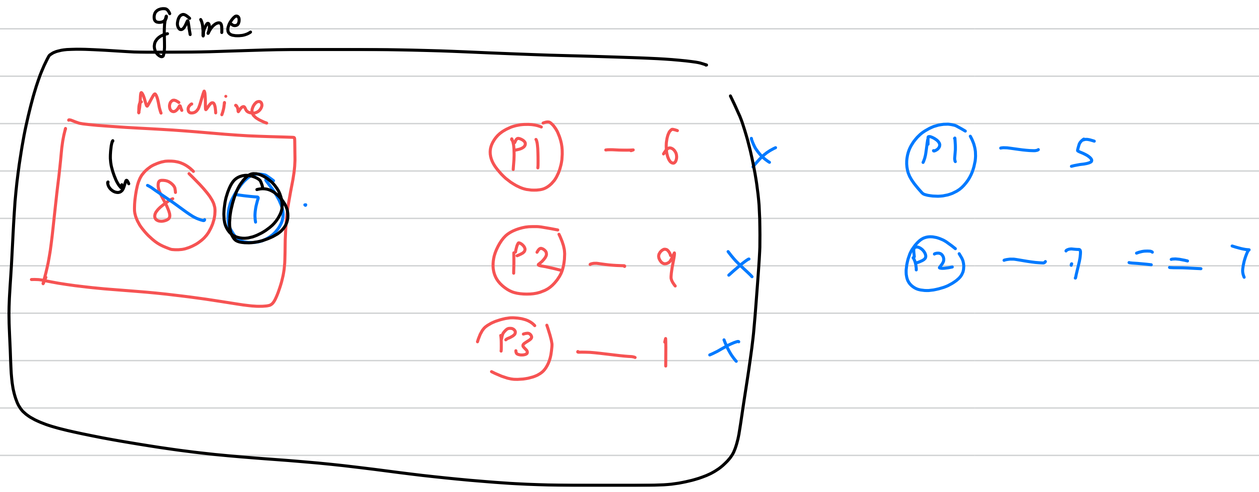
Python

copy obj = copy.deepcopy(obj)

Python : you can't create multiple constructors

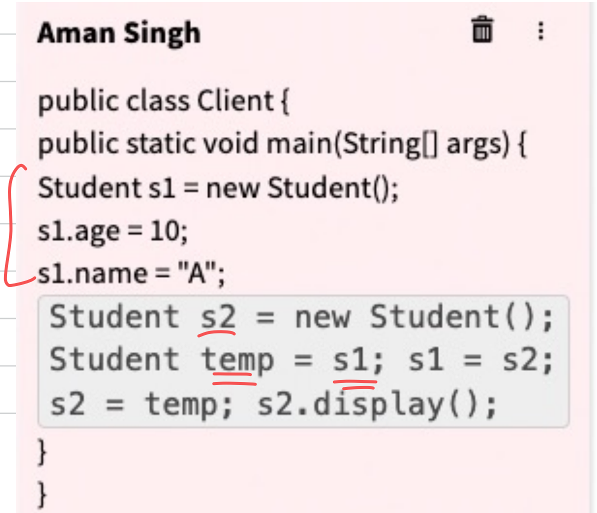
Project

Problem Statement - Create a 3 player game, in which a computer generates a random integer between 1-9. Each player has to make a random guess, guessing the number. The player takes turn in order to make a guess, the player who guesses the number correctly wins the game, if all three players make a wrong guess, the game starts again with computer making a new guess. Think about the entities, and their attributes, and the actions they can perform. Design & execute the game using OOPS principles.



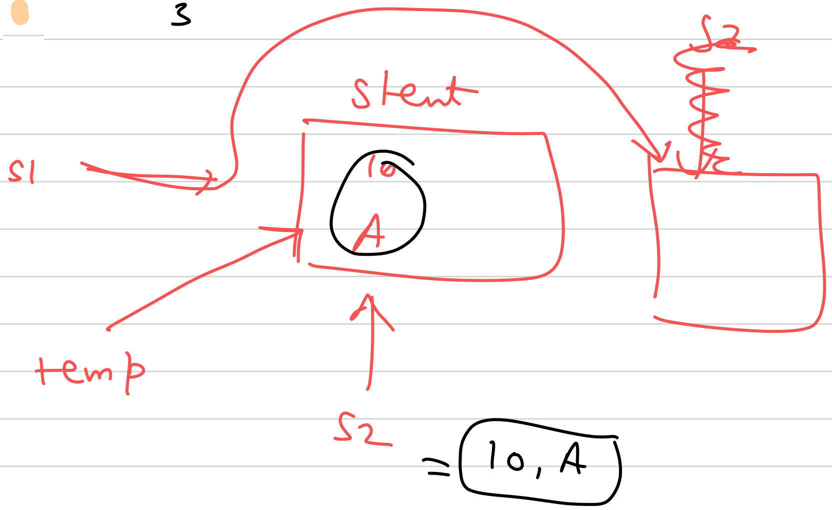
```
Class Game {

    int randomNo,
    Player p1, p2, p3,

    play() {
        ≡
    }
```

```
Class Player {
    String name,
    int guess,

    void makeGuess(){
        guess = someRandomNo( );
    }

}
```



Aman Singh

```
public class Client {
public static void main(String[] args) {
Student s1 = new Student();
s1.age = 10;
s1.name = "A";
Student s2 = new Student();
Student temp = s1; s1 = s2;
s2 = temp; s2.display();
}
}
```



s2
Stent
s1
10
A
temp
s2
= 10, A