FUN CTIONS / METHODS

- · Functions | Methods Introduction.
 - ⇒ a block of code which only runs when it is called.
 - -> it is defined once and reused multiple times.

Syntax: -

access modifier return-type name () {

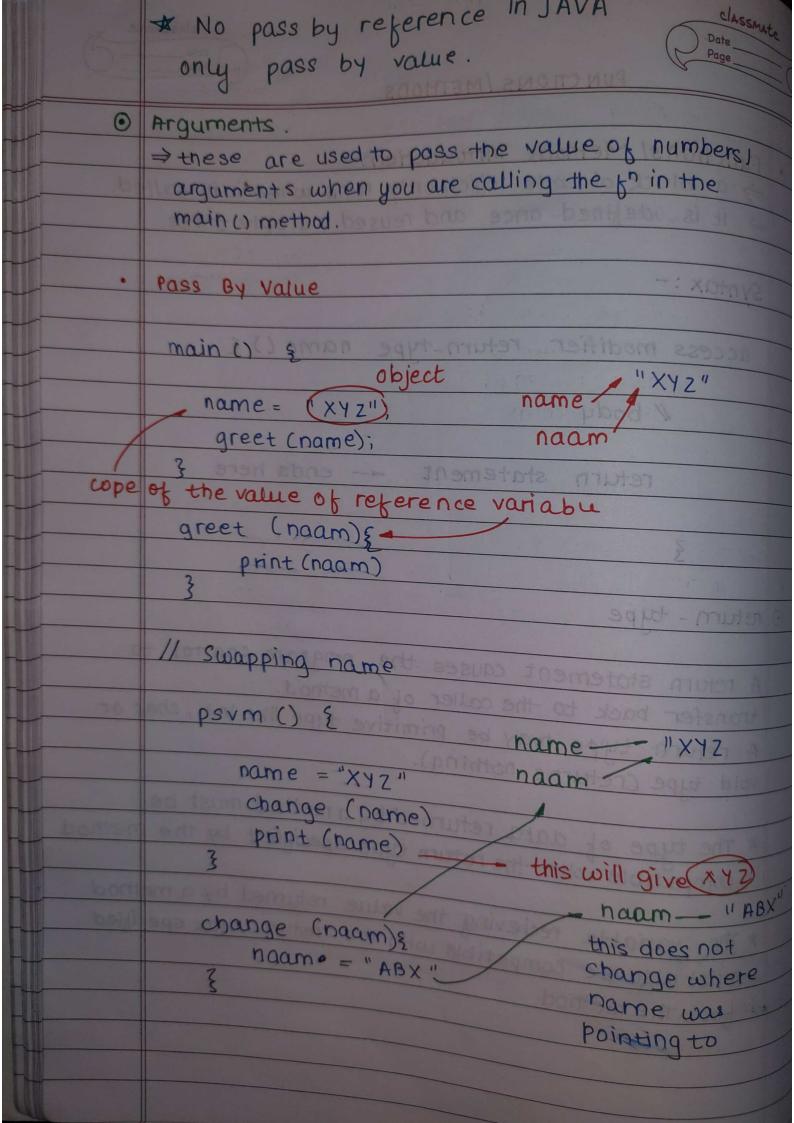
1 body

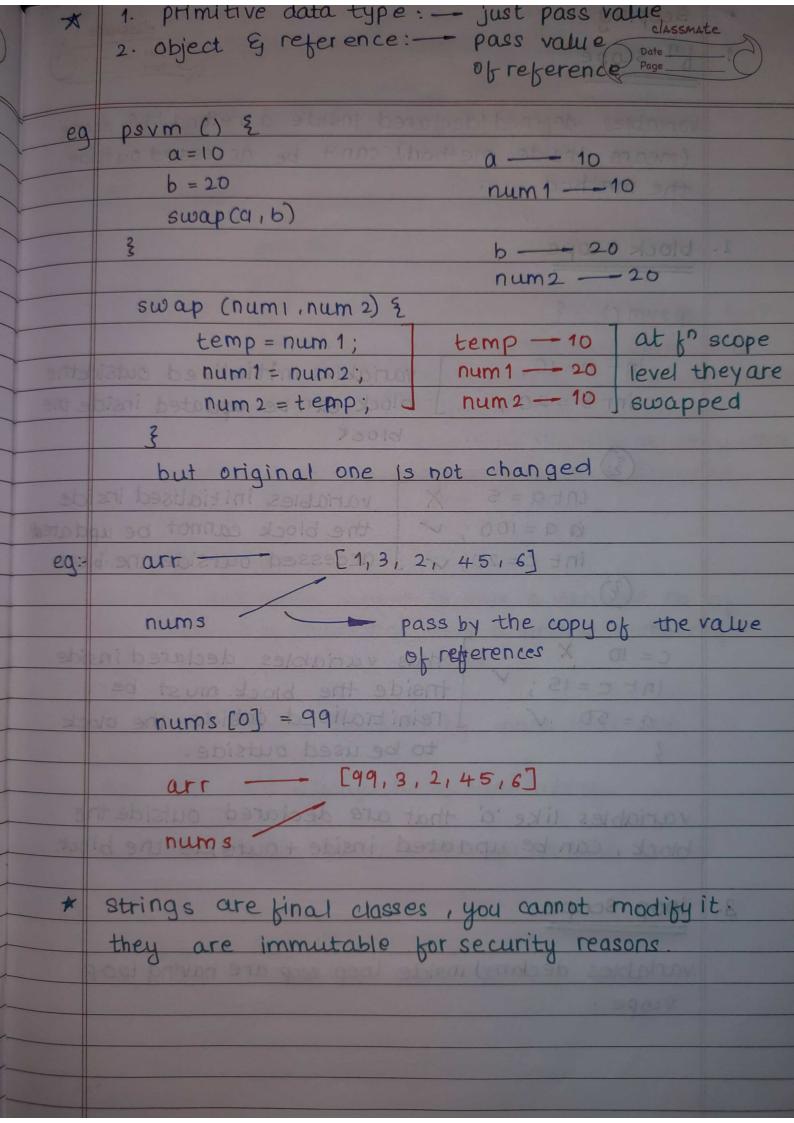
return statement - ends here

3

o return - type

- A return statement causes the program control to transfer back to the caller of a method.
- A return type may be primitive type like int, char or void type (returns nothing).
- * The type of data returned by a method must be compatible with the return type specified by the method
- * The variable recieving the value returned by a method must also be compatible with the return-type specified for the method.





	· Scoping classmate
	1. En scope
	variables defined Ideclared Inside a method I for scope (means Inside method) can't be accessed outside the method.
2	· block scope
9903 PUD Y9	psvm() { int a=10; variables inttialised outsidethe
63	int b = 20; block can be updated inside the
	block
	int a = 5 X variables initialised inside in a = 100; V the block cannot be update int c = 20; V accessed outside the block (3)
Sulpi	nums for pass by the copy of the
	thus variables declared inside int c=15; this variables declared inside a=50 / Teinitialised ontide the block to be used outside. variables like a that are declared outside the block, can be updated inside + outside the block.
3.	loop scope
	variables declared inside loop scop are having loop Scope.

Shadowing practice of using variables with the classmate same name in overlapping scopes Date_Page where the variable in low-level stope overrides the variable of high level scope. eg:- public class Shadowing () { static Int x = 90; psvm () { scope of this local variable that int x; shadows the class variable is not beginning at the point where x = 40; the local scope begins but shadowing begins actually when local variable is declared L- x = 40; variable Arguments. variable arguments is used to take a variable no. of arguments. A method that takes a variable number of arguments is a varargs method. function called be called by more than one argument. the input will always be of type array. variable length arguments always at end.

function overloading Method overloading Two ore more functions can exist of the same name if the parameters are different. fun (int a) { 11 code This is allowed. having different arguments with fun (stringa) { same method name. => At compile time, it decides which for to run. tariable Arguments: