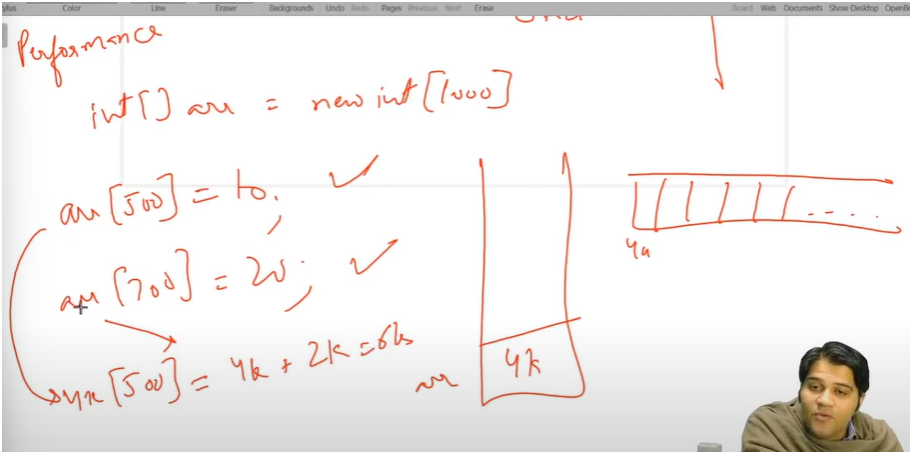


**🧠 Memory Areas: Stack vs Heap**

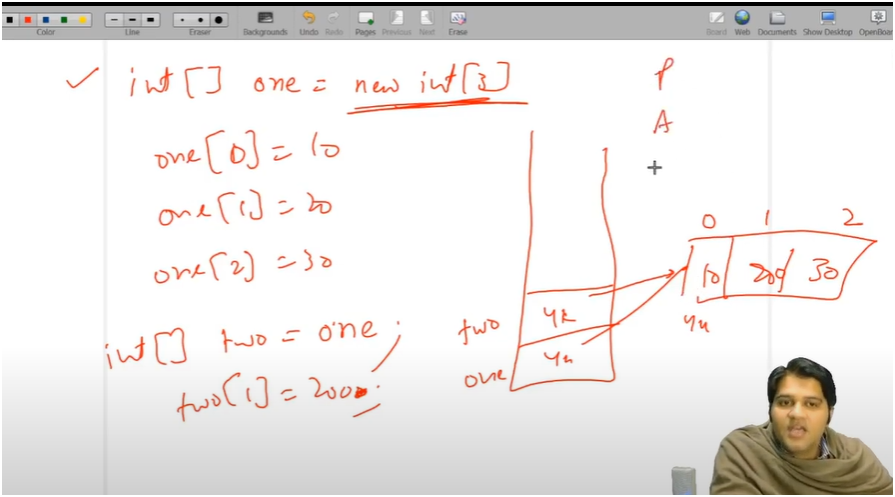
* **Stack**:
  + Stores **starting address** of arrays or objects.
  + Used for **method calls and local variables**.
* **Heap**:
  + Stores **actual data** (e.g., array elements).
  + Memory is allocated **dynamically** at runtime.



**📌PERFORMANCE**

**Array Example:**

* Suppose arr starts at address 4K.
* arr[500] = value means:
  + Address calculation: 4K + 500 × 4 = 6K  
    (since each int is **4 bytes**).
  + So, the value is stored at memory location 6K.



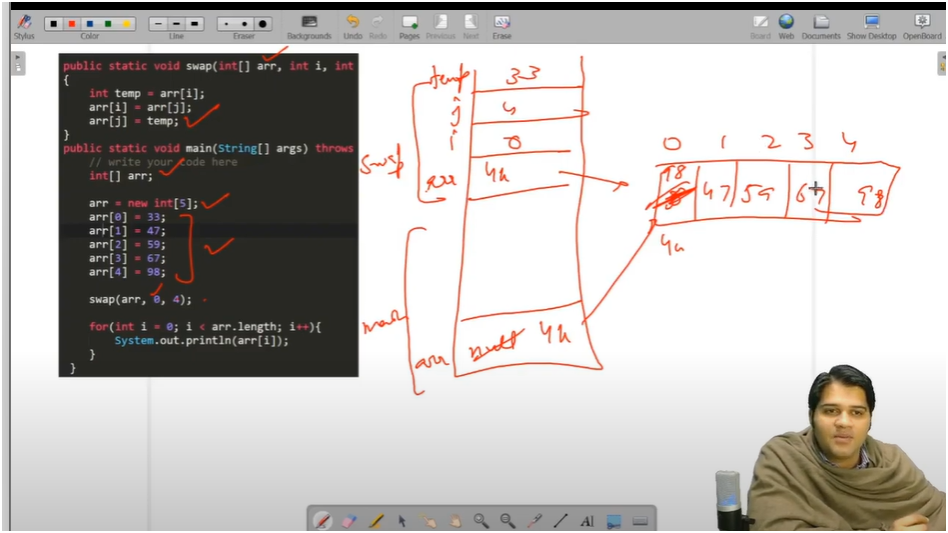
**📥 Assignment Types in Java**

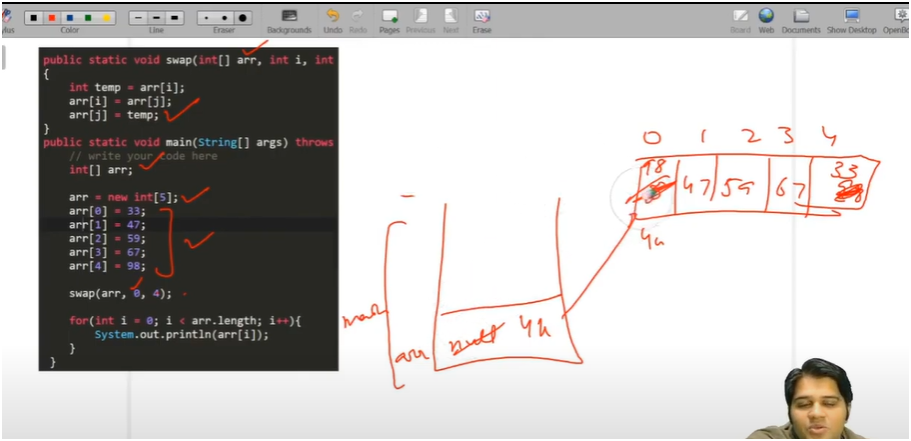
**1. ✅ Deep Copy**

* Creates a **new array** in heap memory.
* Stack stores the **new starting address**.
* **No changes** in the original array affect the new one.

**2. ⚠️ Shallow Copy**

* Stack stores the **same address** as the original array.
* Any changes made to the new array will also reflect in the original array.





**🔁 Function Termination Behavior**

* Once a function finishes executing:
  + **All its local variables are destroyed** (stack frame is cleared).
* Behavior with arrays:
  + **Shallow Copy**: Changes remain visible outside the function.
  + **Deep Copy**: Changes are local to the function and do **not affect** the original array.
* **STACK** 
  + STORES STARTING ADRESS
* **HEAP** 
  + STORES ACTUAL VALUES
* **DEEP COPY**
  + STORES NEW ARRAY ADDRESS
  + CHANGES NOT AFFECTED
* **SHALLOW COPY**
  + STORES EXISTING ADDRESS OF ARRAY
  + CHANGES AFFECTED