RUST

#Introduction : Rust

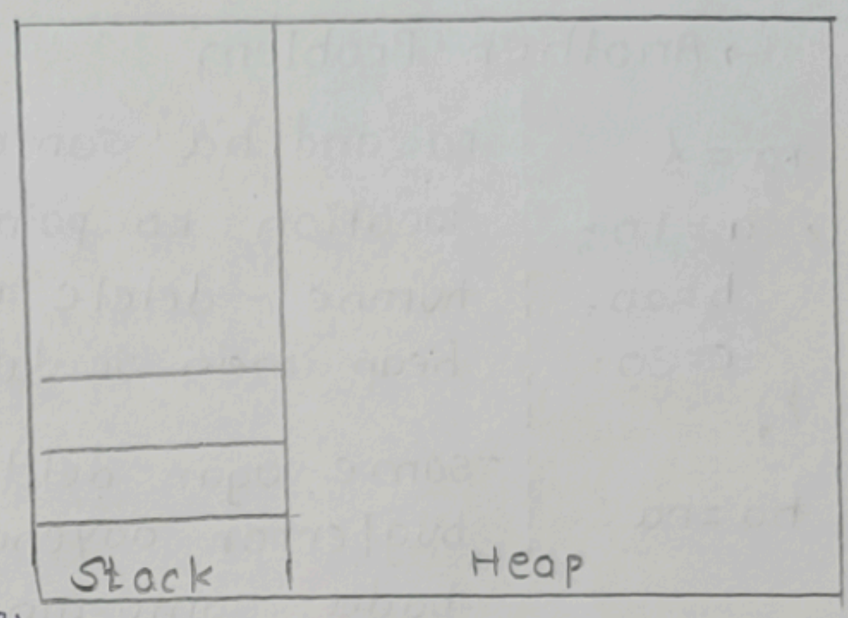
◉Why do we need Rust when we have - C, C++, Java, Python, Js, etc.

1. Why we call C/C++ faster than Java, Python, JS, etc -

→ We all know how dynamic allocation works in C and C++

↳they have new / delete keyword

allocation deallocation



- If we use these keywords : new we will get memory allocated in Heap.

- Now, we know program runs in our RAM and each program is allocated some memory if we use more overflow condition occurs.

- Now, whenever we allocate our memory to heap using new keyword in C/C++, it do not automatically gets deleted. We need to use delete keyword to release the memory in C/C++.

- But we have heard of term - garbage collector in other language like Java, Js, Python.

- Garbage Collector ?
  ⇒ Jaise hum unwanted images delete kar dete hai from gallery waise hi garbage collector unused memory locations ko delele kar deta hai from heap.

- How does Garbage collector do that ?
  ⇒ It stops the running program, checks for garbage values deletes it if found and reruns or continues running the program.

Ans : As Garbage collector pauses running code for some duration frequently, C/C++ becomes faster due to absence of garbage collector

⇒ NOW Rusts uses advantages of both side C/C++ & JS, Java
↳ it will tell us how memory allocation & deallocation hoga

- Rust do not give power to user to allocate and deallocate memory just like in c/c++.

Why ?

↳ Agar humne bahoot khrb quality ka code likha hai - no proper deallocation toh user jaise hi humara application run karega - uska saara os ka space application use karlenga leading to crash.

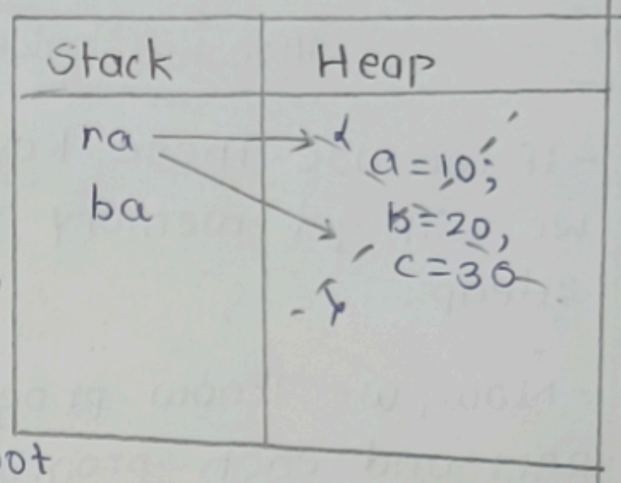
↳ Another Problem

```
ra = {
    a = 10;
    b = 20,
    c = 30.
};

ba = ra
```

ra and ba same memory location ko point karega & if humne - delete ra kiya toh heap mein se data delete ho jayega

- same agar delete ba kiya toh bug/error aayenga and fir bahoot bada confusion hoga.

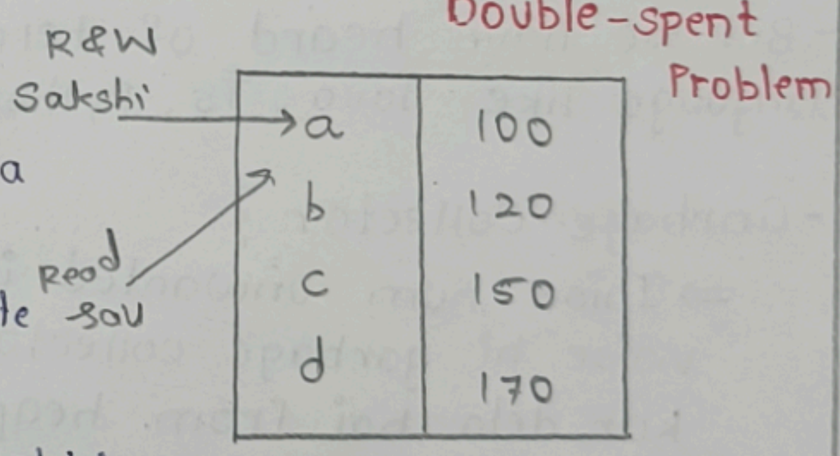


↳ Memory management bahoot hard hai c/c++ mein

- Rust ne kaha agar mein — i) fast
ii) no memory mgmt headache
kitna badiyaa hoga

Rust - Ownership Concept

- Sakshi ke paas power hai to read & write data and suppose Sau and Avi data read kr rhe toh unhe dirty read wali problem aayengi coz sakshi can manipulate data



Double-spent Problem

- Jab Sakshi data manipulate kr rhi thi tabhi Avi ne bhi kiya toh values galat aa skti suppose they both read value as 100 & added 10 ⟹ finalt 110 jbki 120 hona toh

- So Rust restricts this things i.e. ownership : agar owner ke alawa bhi koi read kar sakta hai toh owner write ops nhi kar sakta.

- Only one owner i.e. ek hi person read & write kar sakta
- Multiple reader allowed but owner can't manipulate data.

⊙ Ownership Transfer

1) Only Read (multiple persons ko de sakte)
2) Read and Write (only one person)

Few Doubts asked

1) Ownership temprorary hoti hai yaa permanent?

⇒ Code part mein dekhenge

2) Read and Write ownership transfer ke baad koi read kar sakta hai?

⇒ No

3) Ownership can be given back?

⇒ Haa, agar woh chahe toh ownership transfer back hoti hai

4) Read ownership dene ke baad real owner manipulate kar sakta hai data?

⇒ NO --- (main concept)

5) Read ownership agar wapas aa gayi toh kya owner write kar sakta hai?

⇒ Yes, agar sbse read ownership aa gayi toh tabhi only

#Inshort, koi bhi data manipulate nhi kar sakta if someone is reading data.