what is the problem EJB, RMI on typical Garbage collection process ?

- These legacy frameworks doesn't have any intelligence where/when object is ready to destroy.
- In applications developed using legacy frameworks, Object is created by developer
- so object management --> creating, managing state, destruction --> all things are take care by the developer.
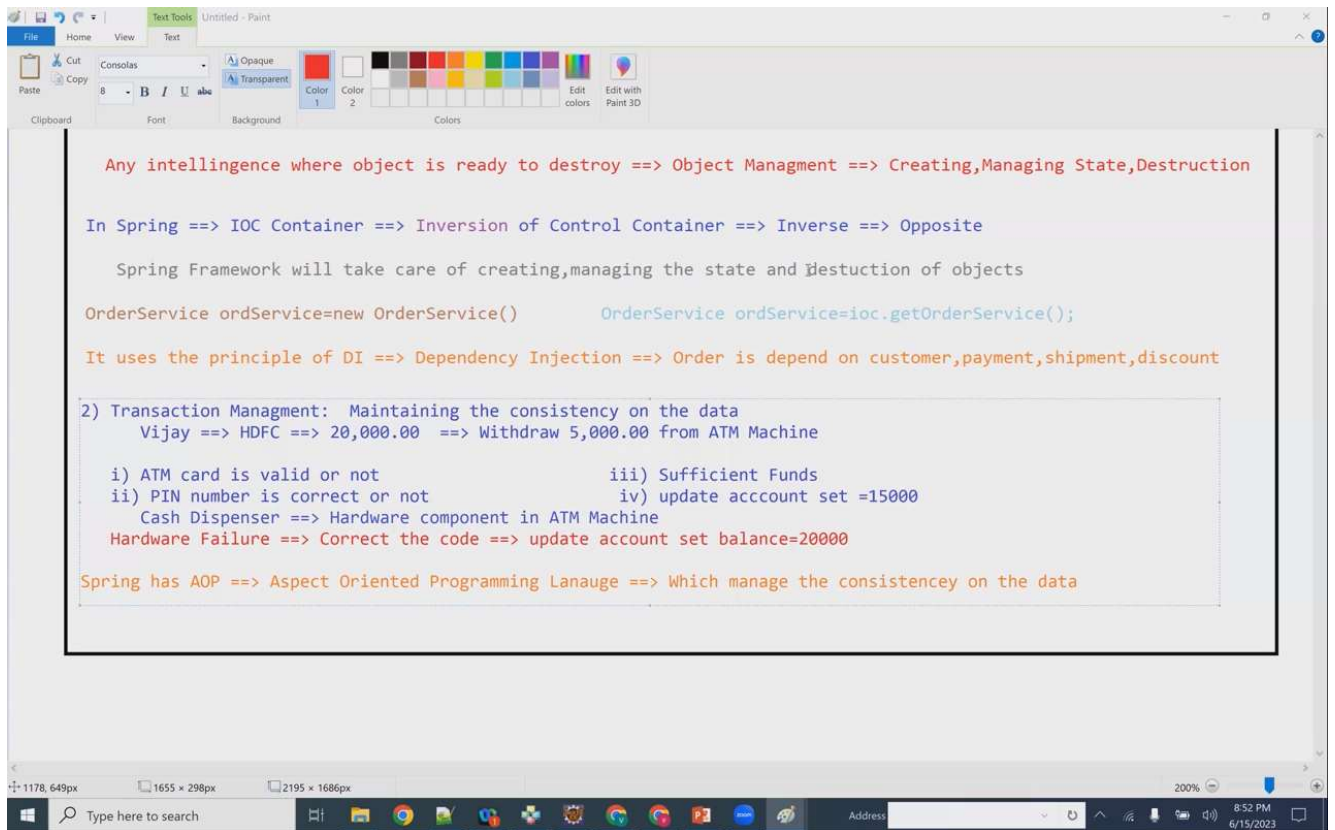
How Spring is solving the above problem?

In Spring,
- it has the concept called IoC Container
- Inversion of Control Container
- Creating the object, Managing state of the object, destruction of the object is taken care by the Spring Framework
- It uses the principle of DI (Dependency Injection)
    - Ex: Order is depend on Customer, Payment, Shipment, discount.

2. No Framework Level Support for Transaction Management
- Maintaining the consistency on the data.

```
    Any intellingence where object is ready to destroy ==> Object Managment ==> Creating,Managing State,Destruction


 In Spring ==> IOC Container ==> Inversion of Control Container ==> Inverse ==> Opposite

    Spring Framework will take care of creating,managing the state and Destuction of objects

OrderService ordService=new OrderService()          OrderService ordService=ioc.getOrderService();

It uses the principle of DI ==> Dependency Injection ==> Order is depend on customer,payment,shipment,discount


2) Transaction Managment:  Maintaining the consistency on the data
     Vijay ==> HDFC ==> 20,000.00  ==> Withdraw 5,000.00 from ATM Machine

  i) ATM card is valid or not                    iii) Sufficient Funds
  ii) PIN number is correct or not               iv) update acccount set =15000
     Cash Dispenser ==> Hardware component in ATM Machine
   Hardware Failure ==> Correct the code ==> update account set balance=20000

Spring has AOP ==> Aspect Oriented Programming Lanauge ==> Which manage the consistencey on the data
```

===