OOPS :

1. Data hiding
2. Abstraction
3. Encapsulation
4. Tightly encapsulated class
5. Is-A relationship (Inheritance)
6. Has-A relationship
7. Method Signature
8. Overloading (Polymorphism)
9. Overriding (Polymorphism)
10. Static control flow
11. Instance Control Flow
12. Constructors
13. Coupling
14. Cohesion
15. Type-Casting
16. Data Hiding :

Outside person can’t access our internal data by external person or our internal data should not go out directly without proper validation/access

E x : Providing user name and password , then only user can access his/her bank account online.

At the same time we can’t access others account information.

How to implement Data hiding?

By declaring data variables/member as Private we can achieve Data hiding.

To access private members we need public methods with proper validation.

1. Abstraction :

**Abstraction** is a process of hiding the implementation details from the user. Оnly the functionality will be provided to the user. In **Java**,**abstraction** is achieved using abstract classes and interfaces

Ex : ATM GUI will show the features offering, but not the internal implementation

Advantages :

1. Security – we are hiding internal implementation logic from the user
2. Enhancement – Without effecting outside person we can able to perform any type of changes in our internal system. Hence enhancement will become easy.
3. Improves maintainability of the application

3.Encapsulation :

Encapsulation = Data Hiding + Abstraction. it means if any component is following Abstraction as well as data hiding then it is also following Encapsulation. Encapsulation:It is actually the process of binding data members and methods together in a unit.Each class in java **itself** is an example of encapsulation