9 feb

Ans-1 It's mean Binding the data member and member function into a single unit is called encabsulation. In Java class where Private and abstraction is there then it is called encabsulation. By using Private access modifier then it is called data hiding

Ans-2 Combine the data of our application and its manipulate at one place

Reduce the coupling of modules & increase the conesion inside them.

help to increase the Privacy of data, user can't access store value for all of the variables

Ans-3 Setter :- It is used to set the value to the instance variable.

- In setter there is ~~no~~ void return type.
- It should be public
- Don't have any argument

Getter :- It is used to get the value from the instance variable.

- Return type should be int
- It should be public
- ~~Don't~~ have any argument.

```
Class a{
    int age;

  Public void setage(int age){
  this.age =age;
      }
    Public int getage( ){
        return age;
          }
  Class b{
      Public static void main(String[] args){
          a a, = new a()
            a,. setage(10);
            a,. getage( );
```

Ans-4 This keyword is used when local & static variable have same name. This problem is called shadowing problem. In this JVM will give most preference to the local variable and result will be name clash. So here this. is used with instance variable so JVM will know that is instance variable and there is no name clash. Est

```
class a
  String name;
      Public ~~void~~ String show(String name){
             this.name = name;
              return name;
             }

      Class b{
        Public static void main(String[] args){
          a a = new a( );
          a,. show("sachin");
          }
        }
```

Ans 5

- Class maintain data member and member function
- It also provide the hiding data to outside world and provide set of services.
- with private there is no direct access

Ans 6

By doing our variable private so that outer world can't access the data and using that private variable into method so that we can call and the data will also safe.

```
Eg. Class a {
        Private age;
    Public ~~int~~ Show (int age) {
        this.age = age;
        return age;
        }
    Class b {
        public Static void main (String [] args) {
            a a, = new a();
            a, . Show();
```