Page No.:
Date : / /

Sixth Day

Topics to be covered :-

OOPs — Encapsulation — Access Modifiers

Extras:

Exercises
- Constructors
- Polymorphism
- Inheritance

Encapsulation

Encapsulation in Java is a fundamental concept that combines data and methods into a single unit called class.

↳ It promotes the idea of bundling related data and behaviours together,
And restricting direct access to the data from outside the class.

* Main goal of encapsulation is to provide data hiding and ensure that the internal state of an object is accessed and modified only through controlled Methods.

Page No.:
Date : / /

Key Aspects and benefits of encapsulation

- **Data Hiding :**

> Encapsulation allows you to hide the internal details and implementation of a class, making the internal state inaccessible to other classes.

> The class provides a public interface (public methods) through which other classes can interact with the object and manipulate its state.

- **Data Protection and Integrity :**

> By encapsulating data, you can enforce constraints, validation and business rules within the class Methods.

> This ensures that the data remains in a valid and consistent state and prevents unauthorized modifications.

- **Code Maintainability :**

> Encapsulation ~~prevents~~ promotes code orgainizations and modular design.

> Since the internal details are hidden, changes to the internal implementation of a class don't affect other parts of code that use the class's public interface.

Page No.:
Date : / /

☆ Main part of Encapsulation

- **Access Modifiers**

are keywords used to control the visibility and accessibility of classes, methods and variables.

~there are 4 access modifiers in Java~

1. Public :

- The 'public' modifier is the least restrictive access modifier.
- Members (fields and methods) declared as 'public' can be accessed from any class or package.
- Public members are part of the class's public API and can be used by other classes.

2. Private :

- The 'private' modifier is the most restrictive access modifier.
- Members declared as 'private' can only be accessed within the same class.
- They are not accessible from any other class or package.
- This provides the highest level of encapsulation and data hiding.

Page No.:
Date : / /

3. Protected :

- The 'protected' modifier allows access within the same package and by subclasses (even they are in diff. packages)
- Protected members are not accessible by classes in diff. packages that are not subclasses of the declaring class
- This access modifier is often used when implementing inheritance and providing controlled access to subclasses.

4. Default (package-private) :

- The default (also known as package-private) access modifier is used when no explicit access modifier is specified.
- Members with default access can be accessed within the same package.
- They are not accessible from outside the package, even by subclasses in different packages.