Day-6

Agenda

OOP’S

Collections

OOP’s Concept

Abstraction

Encapsulation

Polymorphism

Inheritance

OOP’s Concept

Abstraction

Hiding internal implementation details & just highlights the set of service what we are offering is called “Abstraction”

Encapsulation

A class said to be highly encapsulated iff every data members declared as private .

OR

Encapsulation data & corresponding methods in to a single modules is called “encapsulation”

OOP’s Concept

Polymorphism

Where the nature of object decides at run time is called polymorphism .

Compile Time

Run Time

OOP’s Concept

Polymorphism

one object is responsible to perform multiple task at run time is called polymorphism .

There are Two types of polymorphism

Compile Time or Method Overloading

Run Time or Method Overriding

What is Covarient Return type

Child return type need not to be same as parent method returns type. Where child classes also allowed .

This is only applicable for object type not for primitive type

Ex- Object -> String , Object -> Integer , Number -> Integer etc…

OOP’s Concept

Compile Time or Method Overloading

Two method are said to be overloaded iff method names are same but arguments are different

Run Time or Method Overriding

Two method are said to be overridden iff method name & signature must be matched .

In overriding return type must be matched , but this rules is applied until 1.4 v. But from 1.5 v onwards Covarient returns types are allowed .

We can not override the final & private type method

We can not override the static method , it is called method hiding .

Difference between Overloading & Overriding :

Property	Overloading	Overriding
1. Method Name	Must be same	Must be same
2. Arguments	Must be different	Must be same
3. Method Signature	No restriction	Must be same
4. Private , Static , & final	Can be overloaded	Can not be overloaded
5. Access Modifers	No restriction	We can not decrease the scope
6.Method resolution	Always take care by the compiler based on reference type	Always take care by the JVM based on runtime
7. Also known as	Compile time , static polymorphism , early binding	Run time , Dynamic polymorphism , late binding .

OOP’s Concept

Inheritance

It is also known as “Is-A relationship ”

By using extends keyword we can implement Is-A Relationship

The main advantage of Is-A relationship is reusability of the code

Has-A Relationship

This is also known as composite or aggregation

There is no specific keywords to implement Has-A relationship . Mostly we use new keywords

The main dis-advantage of Has-A relationship is to increase the dependency between class