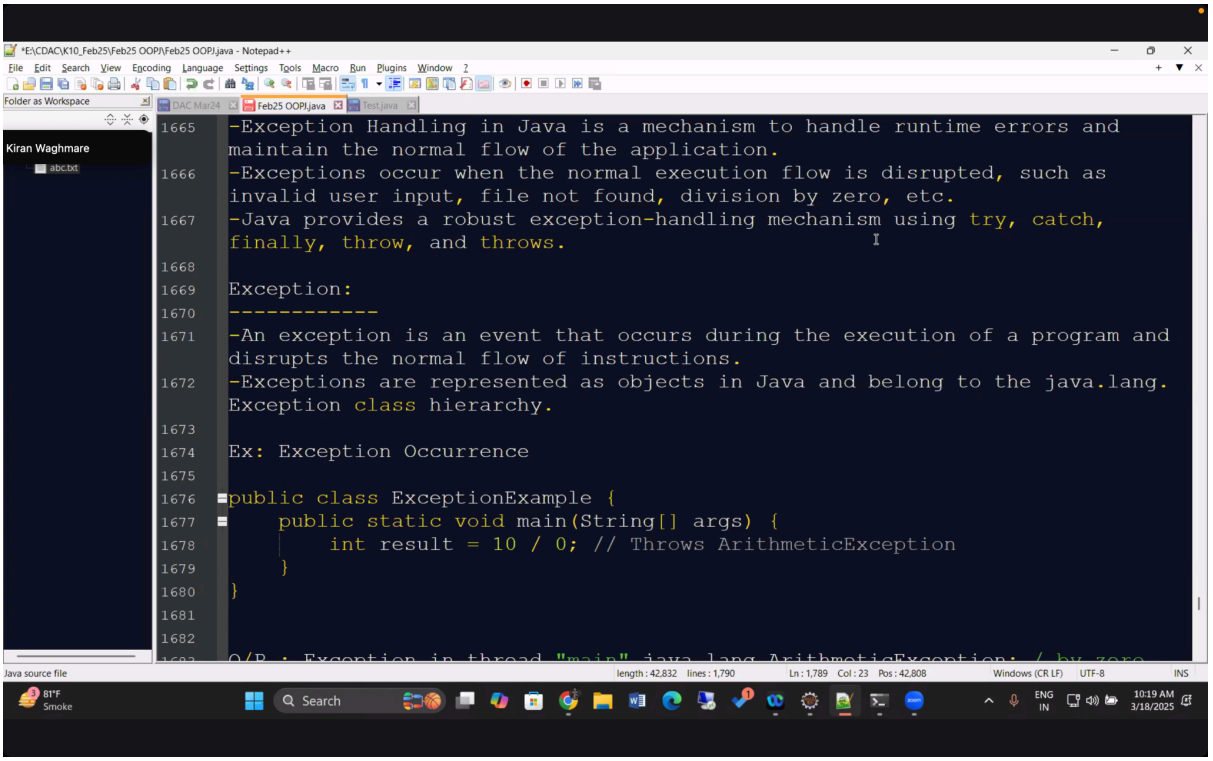
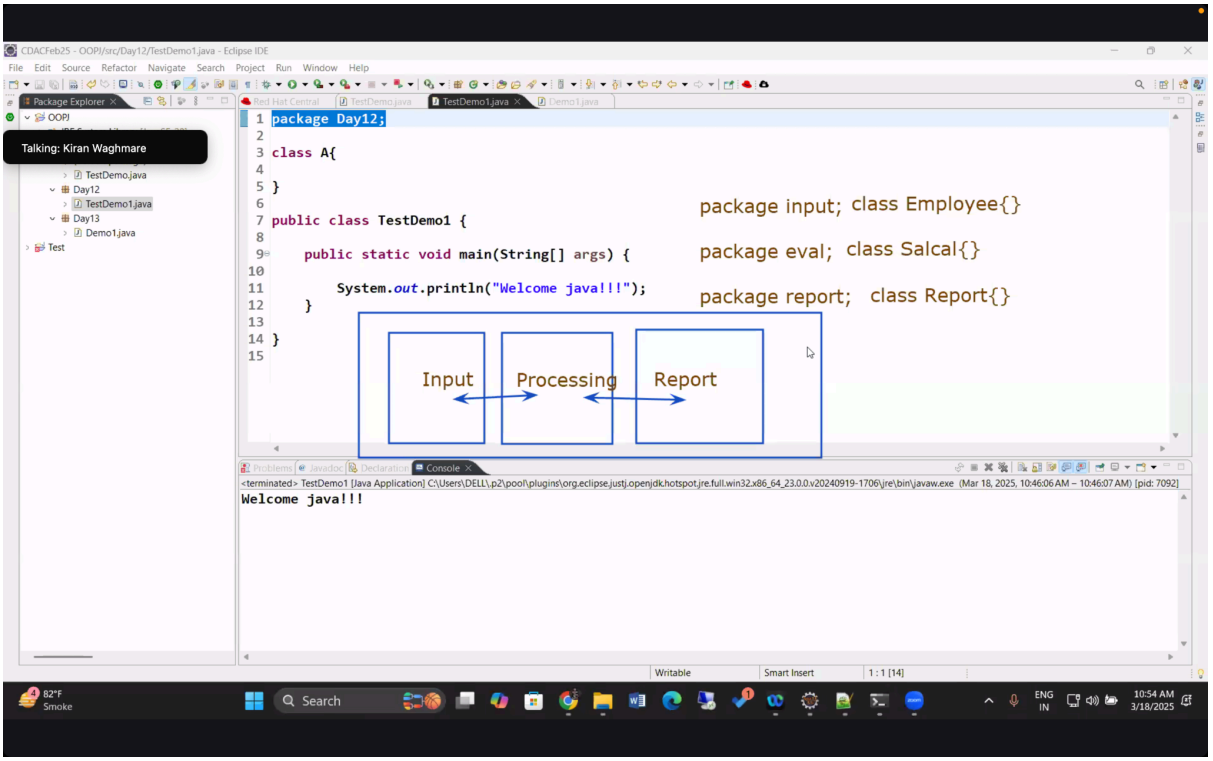
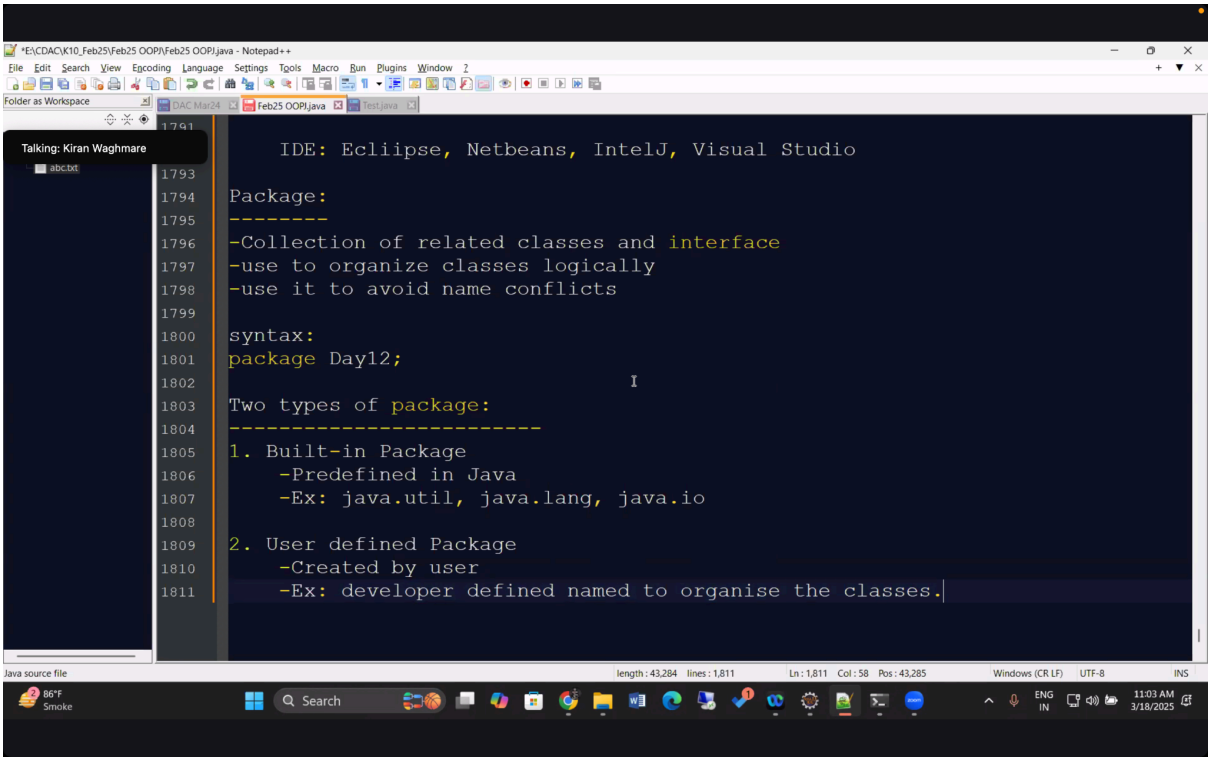
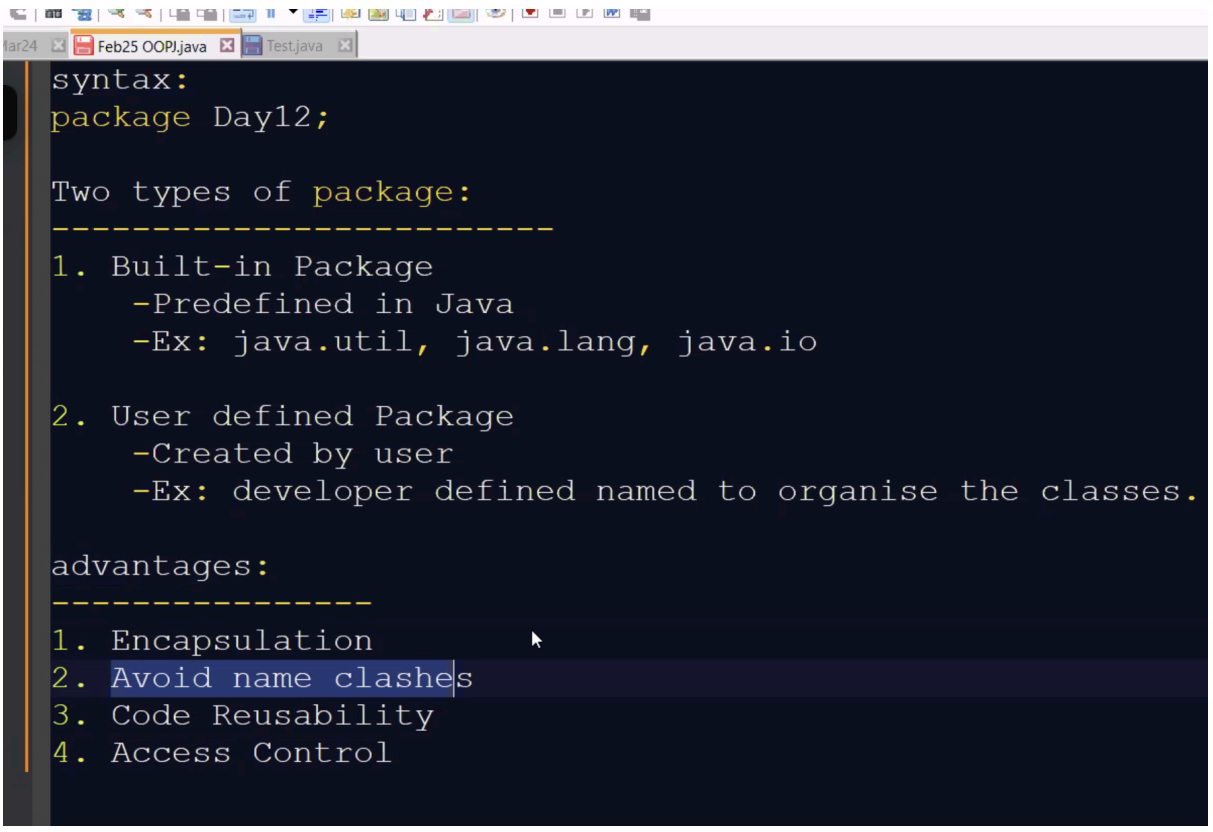
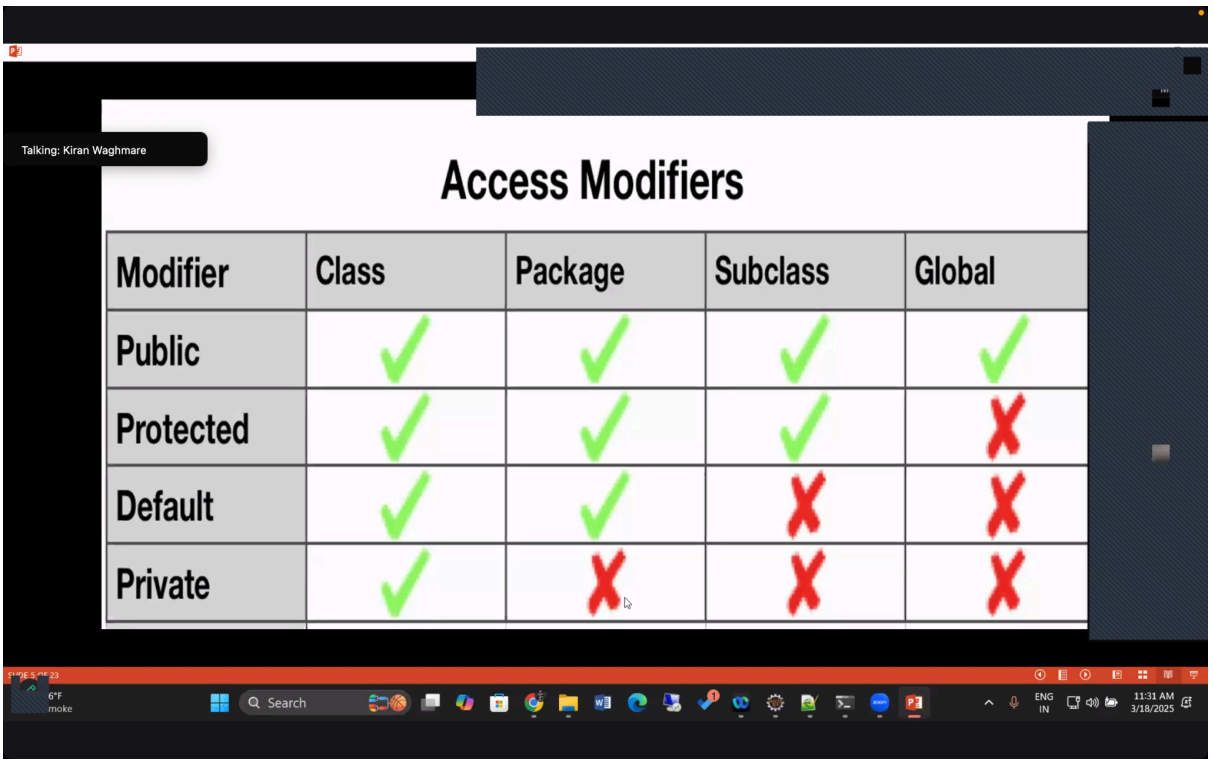
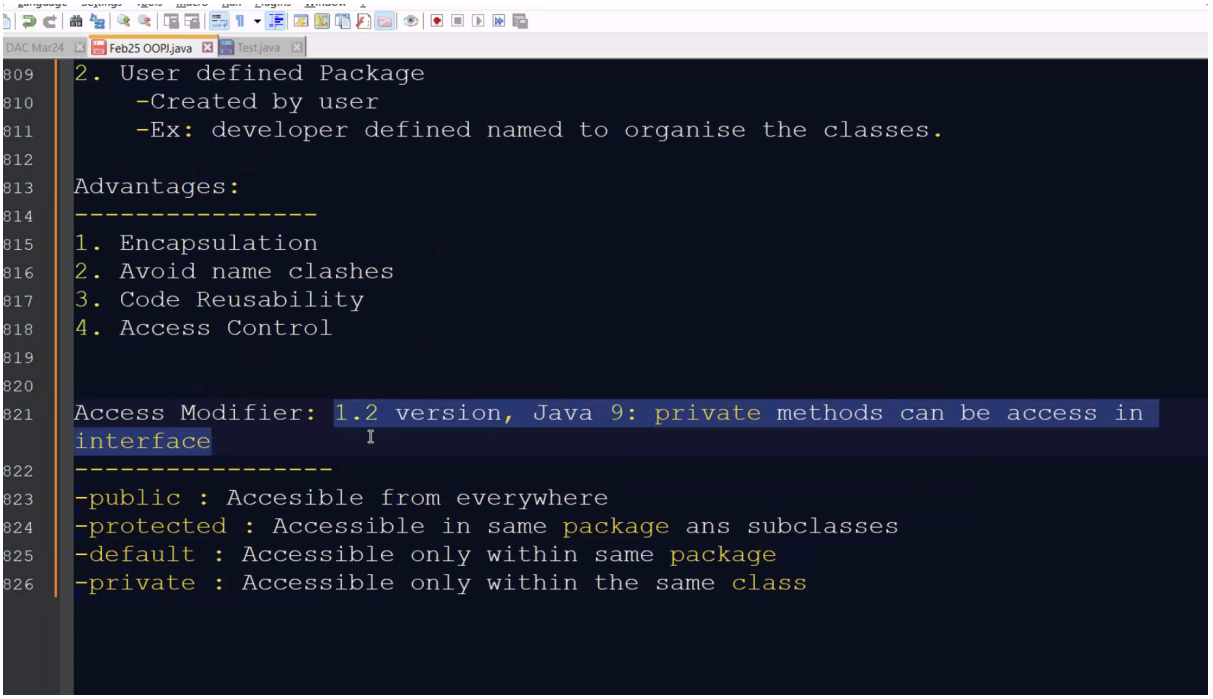

*E:\CDAC\K10_Feb25\Feb25 OOP\Feb25 OOP.java - Notepad++
Kiran Waghmare
abc.txt
1665 -Exception Handling in Java is a mechanism to handle runtime errors and maintain the normal flow of the application.
1666 -Exceptions occur when the normal execution flow is disrupted, such as invalid user input, file not found, division by zero, etc.
1667 -Java provides a robust exception-handling mechanism using try, catch, finally, throw, and throws.
1668
1669 Exception:
1670 ------------
1671 -An exception is an event that occurs during the execution of a program and disrupts the normal flow of instructions.
1672 -Exceptions are represented as objects in Java and belong to the java.lang.Exception class hierarchy.
1673
1674 Ex: Exception Occurrence
1675
1676 public class ExceptionExample {
1677 public static void main(String[] args) {
1678 int result = 10 / 0; // Throws ArithmeticException
1679 }
1680 }
1681
1682
Java source file
length : 42,832 lines : 1,790
Ln : 1,789 Col : 23 Pos : 42,808
Windows (CR LF) UTF-8 INS
81°F Smoke
Search
ENG IN
10:19 AM 3/18/2025

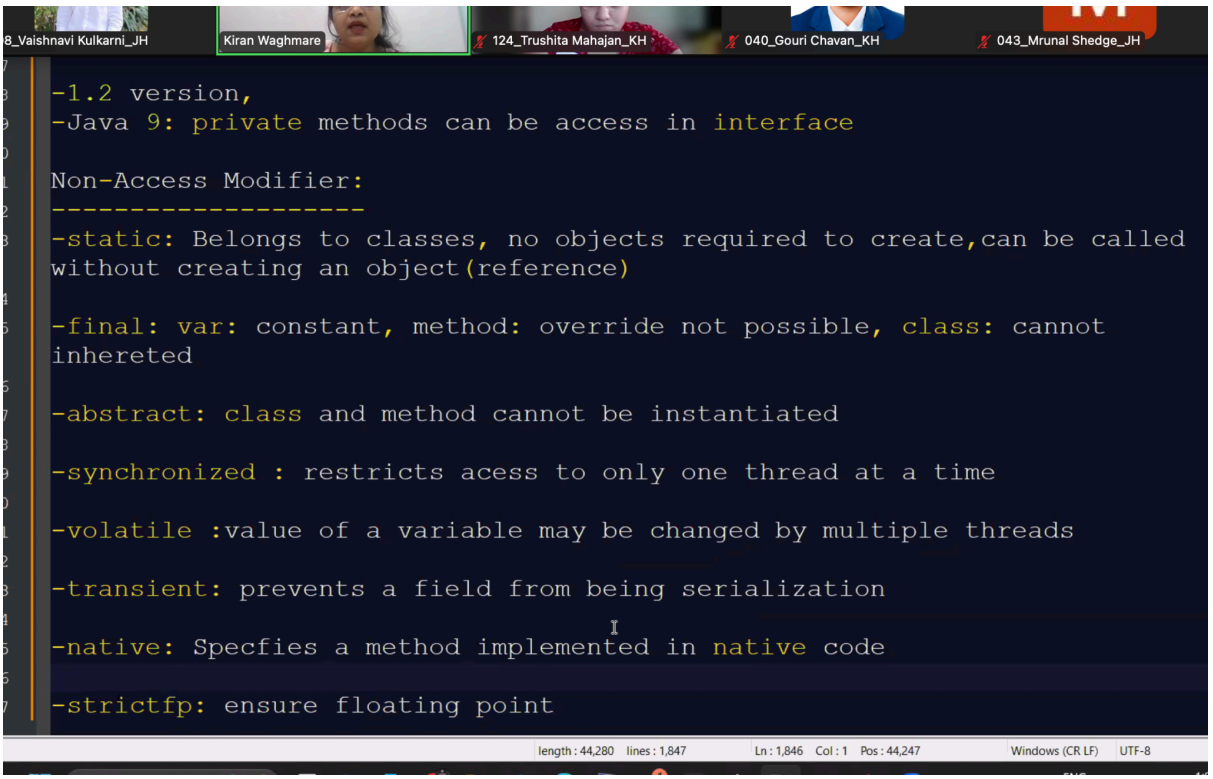

CDACFeb25 - OOP/src/Day12/TestDemo1.java - Eclipse IDE
Talking: Kiran Waghmare
OOP
TestDemo.java
Day12
TestDemo1.java
Day13
Demo1.java
Test
1 package Day12;
2
3 class A{
4
5 }
6
7 public class TestDemo1 {
8
9 public static void main(String[] args) {
10
11 System.out.println("Welcome java!!!");
12 }
13
14 }
15
package input; class Employee{}
package eval; class Salcal{}
package report; class Report{}
Input
Processing
Report
<terminated> TestDemo1 [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\javaw.exe (Mar 18, 2025, 10:46:06 AM – 10:46:07 AM) [pid: 7092]
Welcome java!!!
Writable
Smart Insert
1 : 1 [14]
82°F Smoke
Search
ENG IN
10:54 AM 3/18/2025


*E:\CDAC\K10_Feb25\Feb25 OOPJ\Feb25 OOPJ.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Folder as Workspace
DAC Mar24
Feb25 OOPJ.java
Test.java
Talking: Kiran Waghmare
abc.txt
1791 IDE: Ecliipse, Netbeans, IntelJ, Visual Studio
1793
1794 Package:
1795 --------
1796 -Collection of related classes and interface
1797 -use to organize classes logically
1798 -use it to avoid name conflicts
1799
1800 syntax:
1801 package Day12;
1802
1803 Two types of package:
1804 ------------------------
1805 1. Built-in Package
1806 -Predefined in Java
1807 -Ex: java.util, java.lang, java.io
1808
1809 2. User defined Package
1810 -Created by user
1811 -Ex: developer defined named to organise the classes.
Java source file
length : 43,284 lines : 1,811
Ln : 1,811 Col : 58 Pos : 43,285
Windows (CR LF) UTF-8 INS
86°F Smoke
Search
ENG IN
11:03 AM 3/18/2025

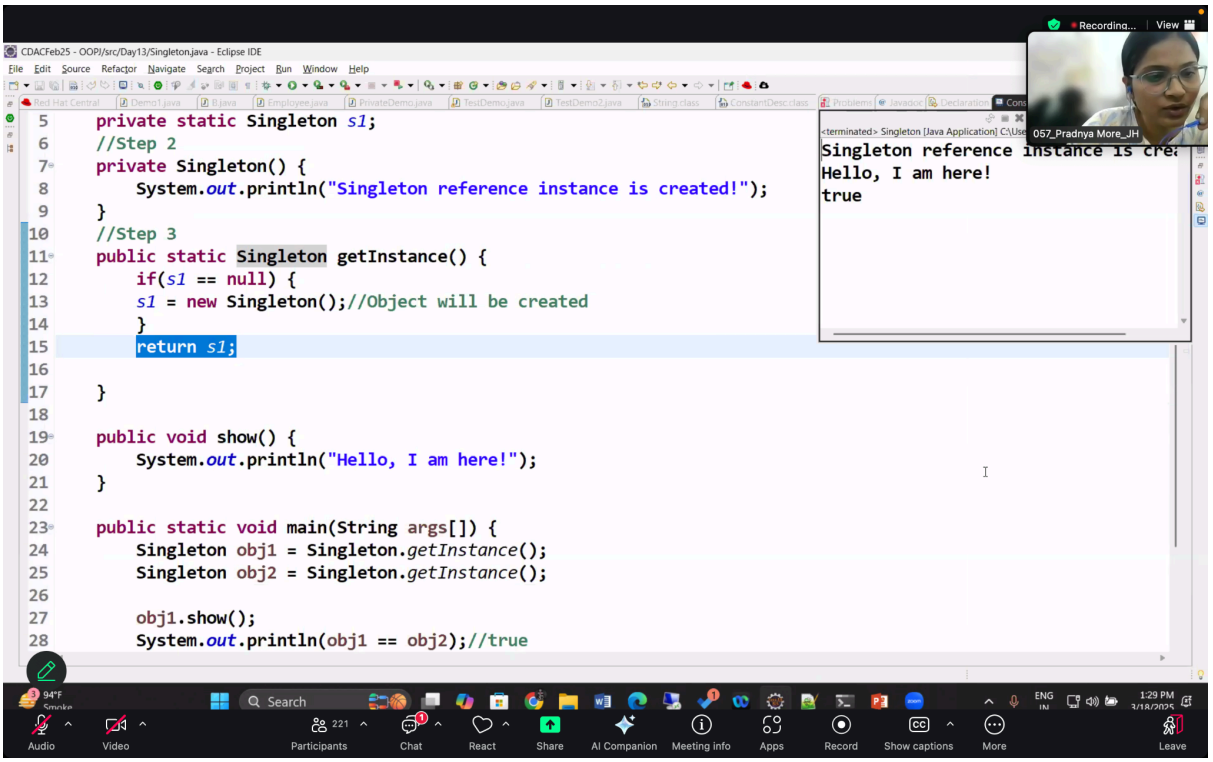

Feb25 OOPJ.java
Test.java
syntax:
package Day12;
Two types of package:
1. Built-in Package
-Predefined in Java
-Ex: java.util, java.lang, java.io
2. User defined Package
-Created by user
-Ex: developer defined named to organise the classes.
advantages:
1. Encapsulation
2. Avoid name clashes
3. Code Reusability
4. Access Control

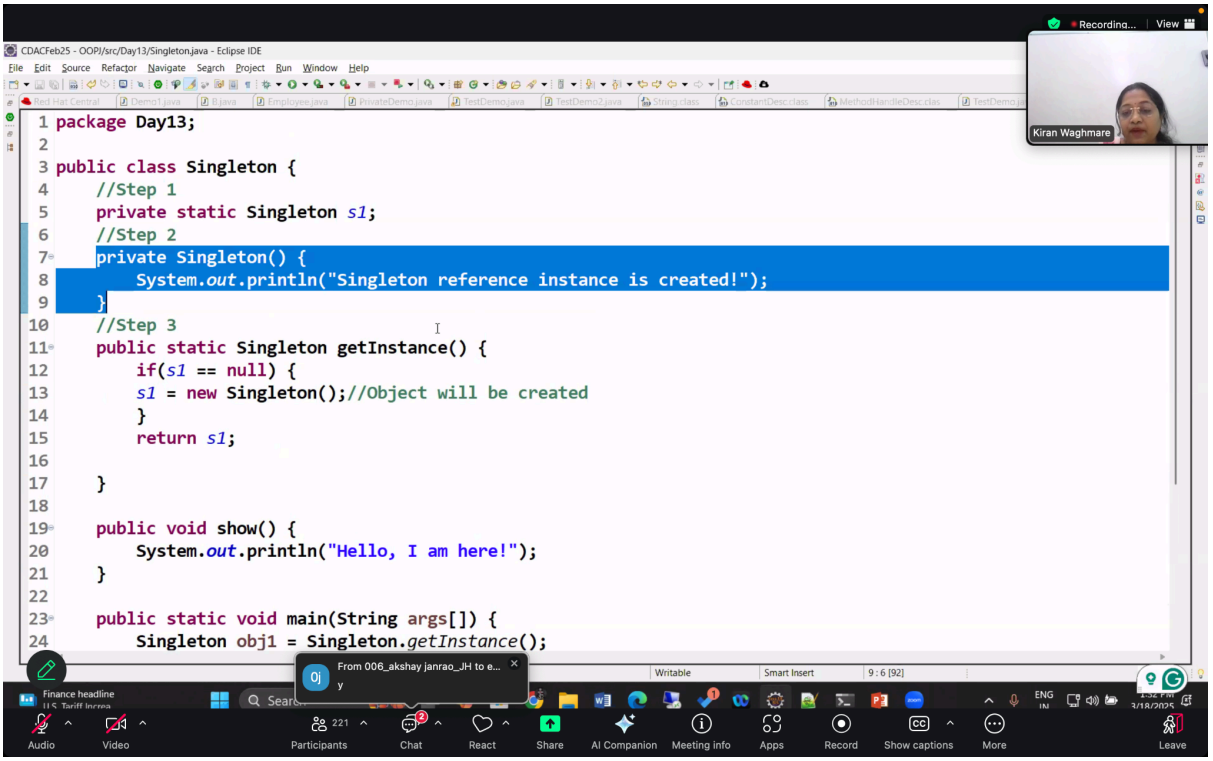

Talking: Kiran Waghmare
Access Modifiers
Modifier
Class
Package
Subclass
Global
Public
Protected
Default
Private
Search
ENG
IN
11:31 AM
3/18/2025

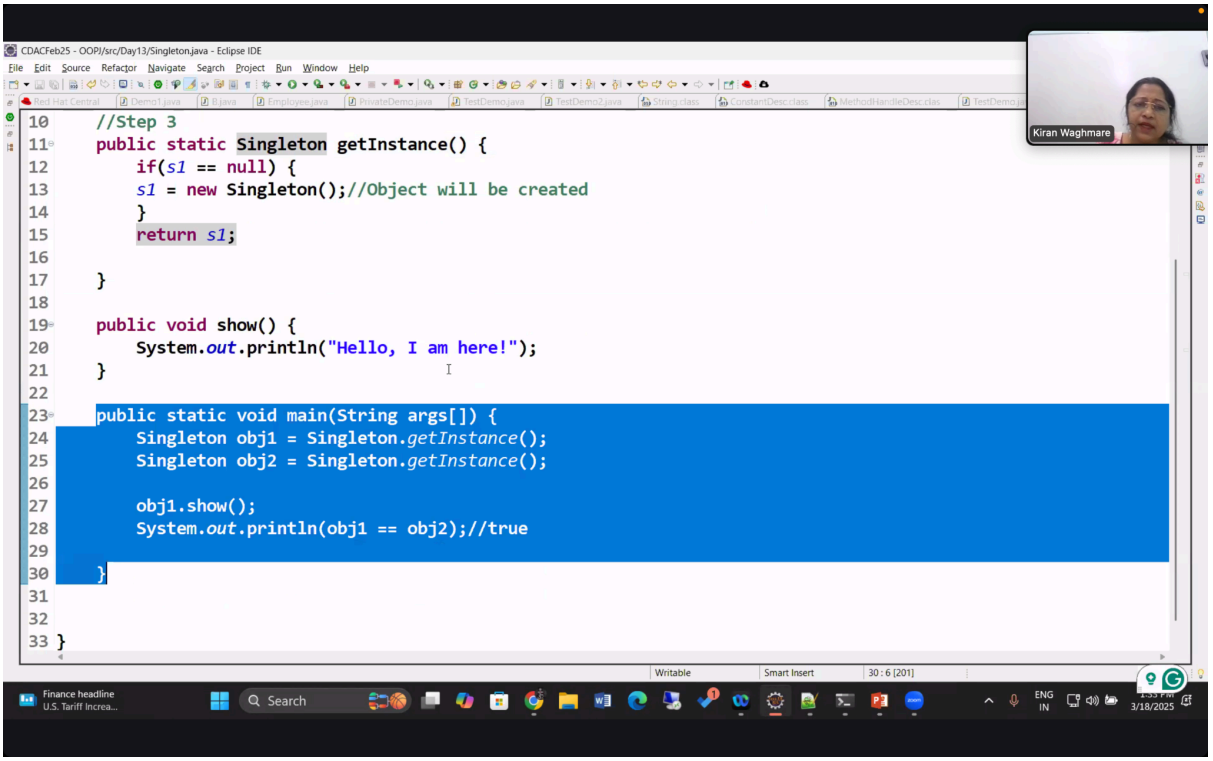

DAC Mar24
Feb25 OOPJ.java
Test.java
2. User defined Package
-Created by user
-Ex: developer defined named to organise the classes.
Advantages:
1. Encapsulation
2. Avoid name clashes
3. Code Reusability
4. Access Control
Access Modifier: 1.2 version, Java 9: private methods can be access in interface
-public : Accesible from everywhere
-protected : Accessible in same package ans subclasses
-default : Accessible only within same package
-private : Accessible only within the same class

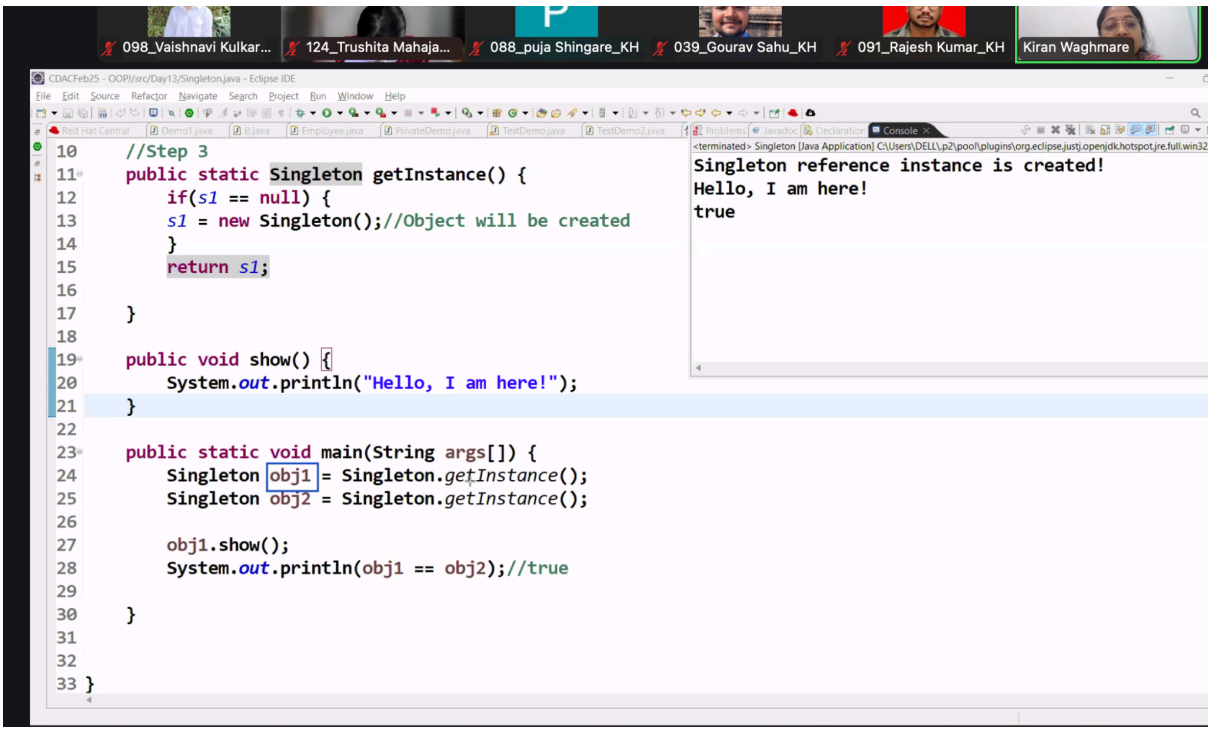

8_Vaishnavi Kulkarni_JH
Kiran Waghmare
124_Trushita Mahajan_KH
040_Gouri Chavan_KH
043_Mrunal Shedge_JH
-1.2 version,
-Java 9: private methods can be access in interface
Non-Access Modifier:

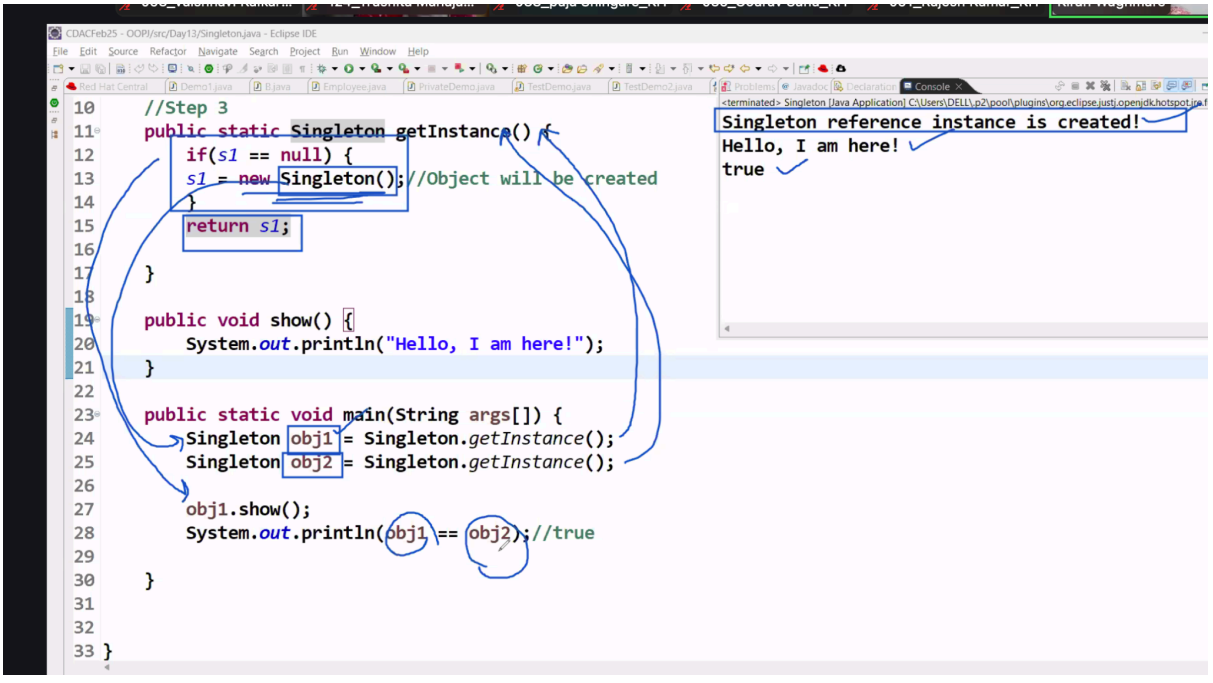
-static: Belongs to classes, no objects required to create,can be called without creating an object(reference)
-final: var: constant, method: override not possible, class: cannot inhereted
-abstract: class and method cannot be instantiated
-synchronized : restricts acess to only one thread at a time
-volatile :value of a variable may be changed by multiple threads
-transient: prevents a field from being serialization
-native: Specfies a method implemented in native code
-strictfp: ensure floating point
length : 44,280 lines : 1,847
Ln : 1,846 Col : 1 Pos : 44,247
Windows (CR LF)
UTF-8

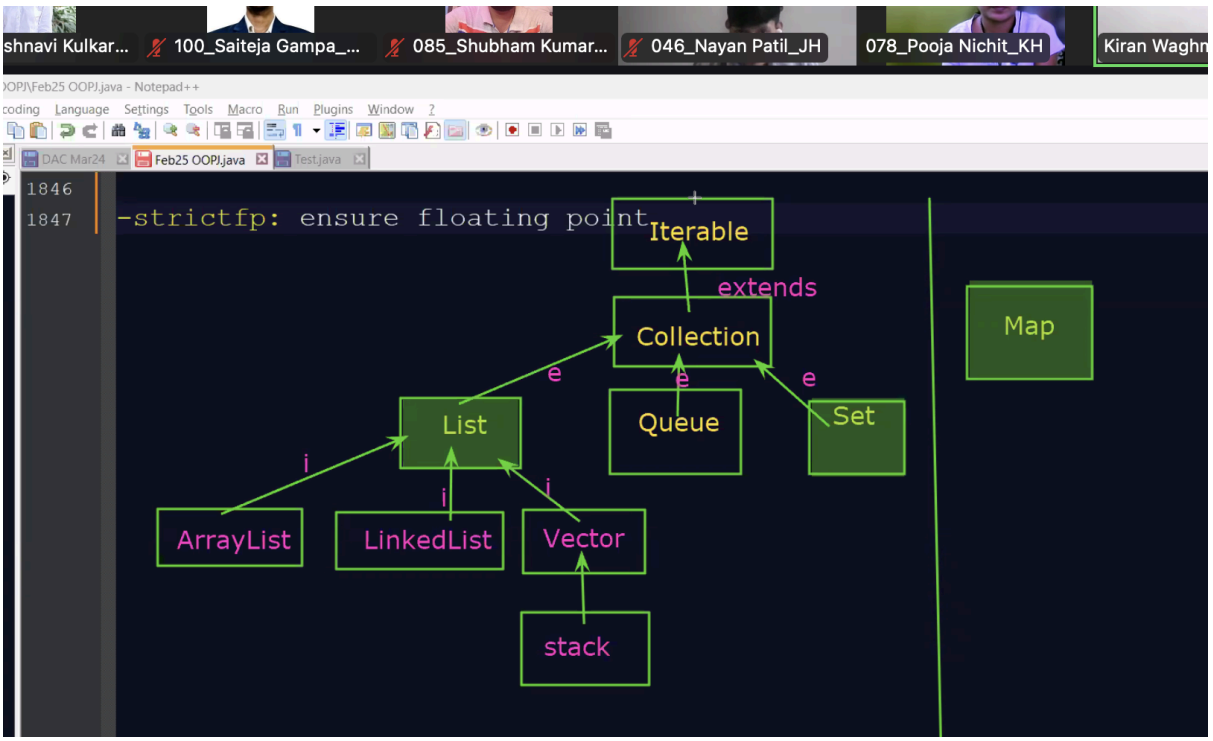

CDACFeb25 - OOP/src/Day13/Singleton.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Recordina...
View
057_Pradnya More_JH
private static Singleton s1;
//Step 2
private Singleton() {
System.out.println("Singleton reference instance is created!");
}
//Step 3
public static Singleton getInstance() {
if(s1 == null) {
s1 = new Singleton();//Object will be created
}
return s1;
}
public void show() {
System.out.println("Hello, I am here!");
}
public static void main(String args[]) {
Singleton obj1 = Singleton.getInstance();
Singleton obj2 = Singleton.getInstance();
obj1.show();
System.out.println(obj1 == obj2);//true
<terminated> Singleton [Java Application] C:\Use
Singleton reference instance is cre
Hello, I am here!
true
Search
Audio
Video
Participants
Chat
React
Share
AI Companion
Meeting info
Apps
Record
Show captions
More
Leave

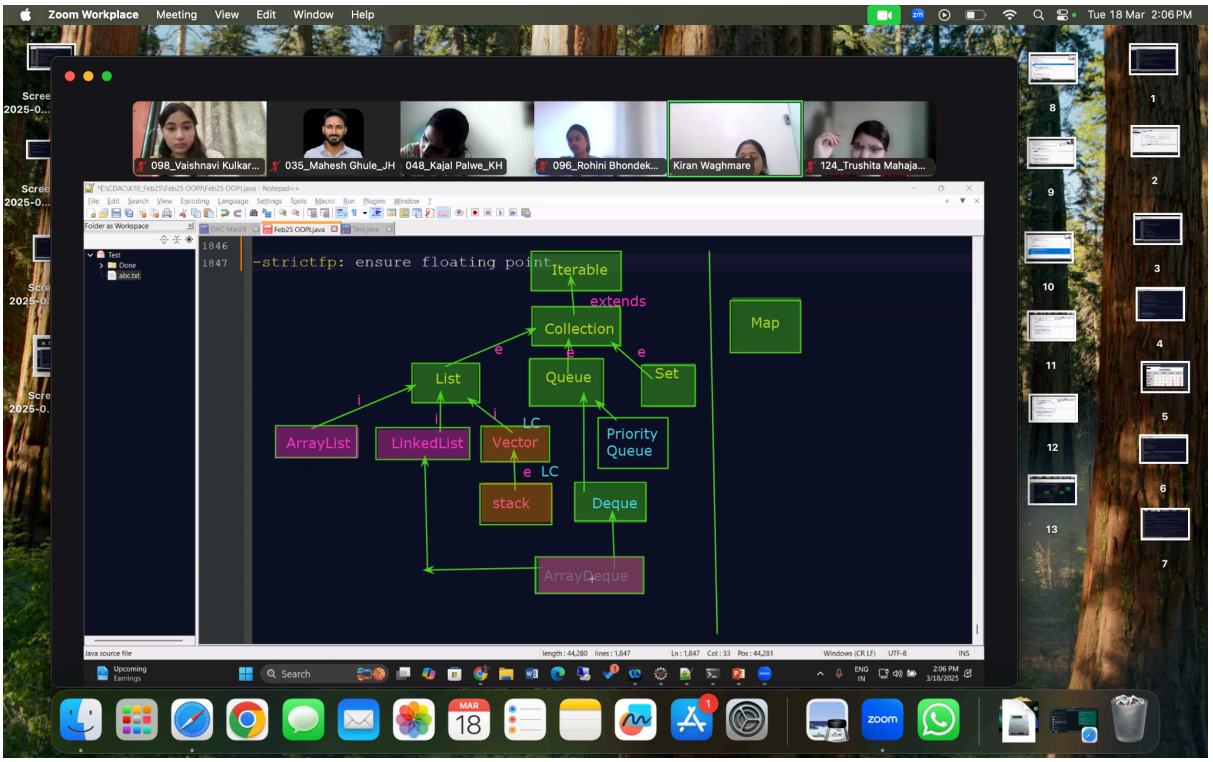

Recordina...
View
Kiran Waghmare
CDACFeb25 - OOPJ/src/Day13/Singleton.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
1 package Day13;
2
3 public class Singleton {
4 //Step 1
5 private static Singleton s1;
6 //Step 2
7 private Singleton() {
8 System.out.println("Singleton reference instance is created!");
9 }
10 //Step 3
11 public static Singleton getInstance() {
12 if(s1 == null) {
13 s1 = new Singleton();//Object will be created
14 }
15 return s1;
16
17 }
18
19 public void show() {
20 System.out.println("Hello, I am here!");
21 }
22
23 public static void main(String args[]) {
24 Singleton obj1 = Singleton.getInstance();
From 006_akshay janrao_JH to e...
y
Writable
Smart Insert
9 : 6 [92]
Finance headline
U.S. Tariff Increa...
Search
ENG
IN
Audio
Video
Participants
Chat
React
Share
AI Companion
Meeting info
Apps
Record
Show captions
More
Leave

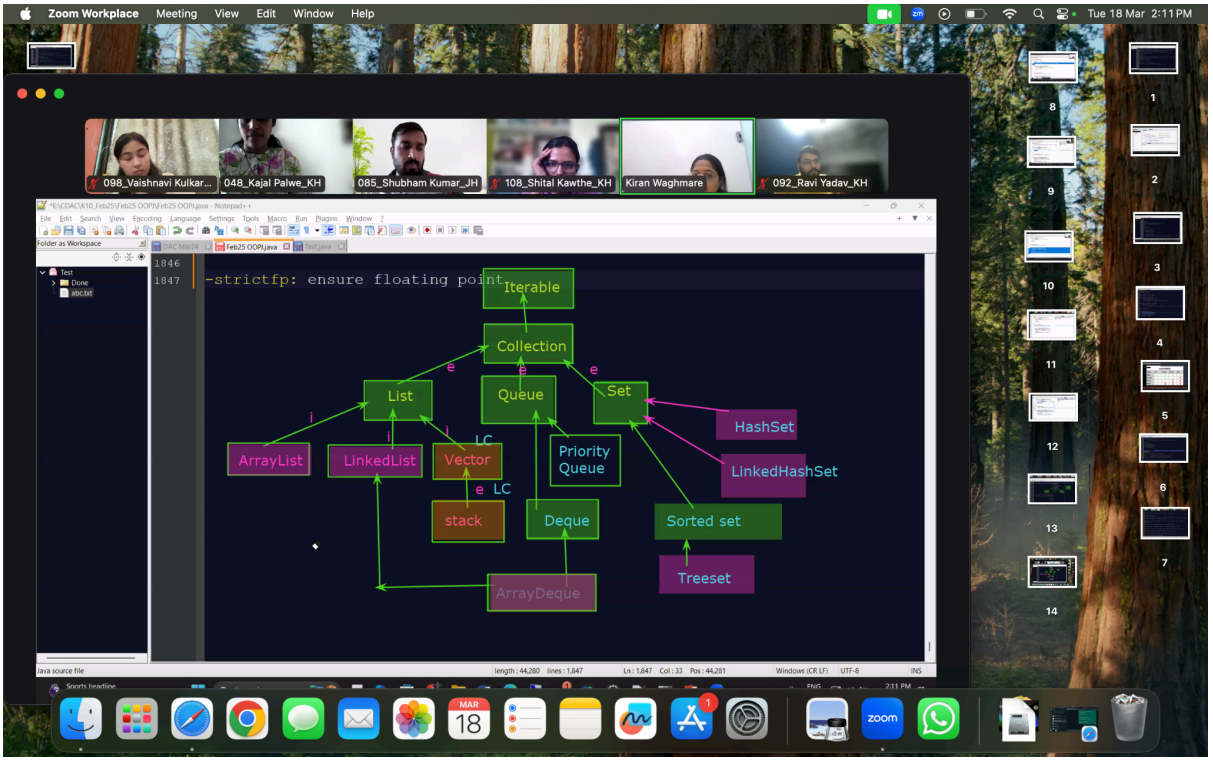

Kiran Waghmare
CDACFeb25 - OOPJ/src/Day13/Singleton.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
10 //Step 3
11 public static Singleton getInstance() {
12 if(s1 == null) {
13 s1 = new Singleton();//Object will be created
14 }
15 return s1;
16
17 }
18
19 public void show() {
20 System.out.println("Hello, I am here!");
21 }
22
23 public static void main(String args[]) {
24 Singleton obj1 = Singleton.getInstance();
25 Singleton obj2 = Singleton.getInstance();
26
27 obj1.show();
28 System.out.println(obj1 == obj2);//true
29
30 }
31
32
33 }
Writable
Smart Insert
30 : 6 [201]
Finance headline
U.S. Tariff Increa...
Search
ENG
IN
3/18/2025

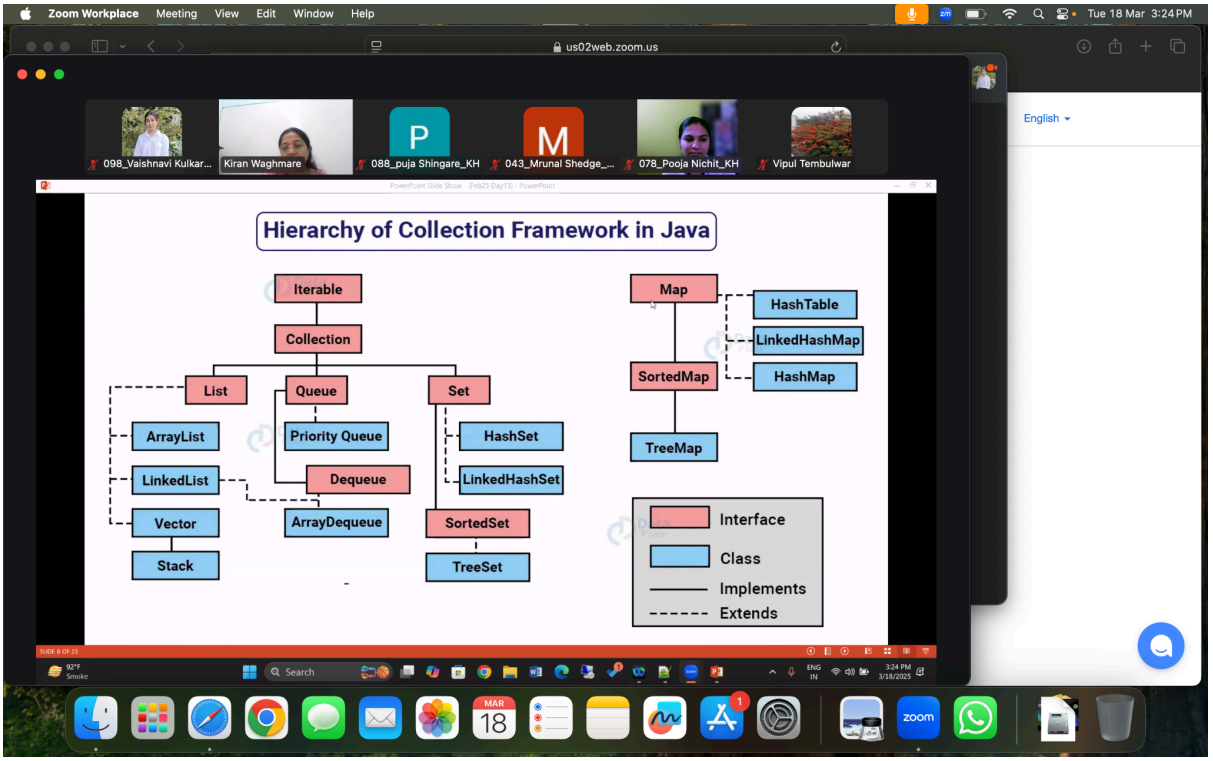

098_Vaishnavi Kulkar...
124_Trushita Mahaja...
088_puja Shingare_KH
039_Gourav Sahu_KH
091_Rajesh Kumar_KH
Kiran Waghmare
CDACFeb25 - OOPJ/src/Day13/Singleton.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Singleton reference instance is created!
Hello, I am here!
true
//Step 3
public static Singleton getInstance() {
if(s1 == null) {
s1 = new Singleton();//Object will be created
}
return s1;
}
public void show() {
System.out.println("Hello, I am here!");
}
public static void main(String args[]) {
Singleton obj1 = Singleton.getInstance();
Singleton obj2 = Singleton.getInstance();
obj1.show();
System.out.println(obj1 == obj2);//true
}
}


CDACFeb25 - OOPJ/src/Day13/Singleton.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Singleton reference instance is created!
Hello, I am here!
true
//Step 3
public static Singleton getInstance() {
if(s1 == null) {
s1 = new Singleton();//Object will be created
}
return s1;
}
public void show() {
System.out.println("Hello, I am here!");
}
public static void main(String args[]) {
Singleton obj1 = Singleton.getInstance();
Singleton obj2 = Singleton.getInstance();
obj1.show();
System.out.println(obj1 == obj2);//true
}
}


shnavi Kulkar...
100_Saiteja Gampa_...
085_Shubham Kumar...
046_Nayan Patil_JH
078_Pooja Nichit_KH
Kiran Waghm
Feb25 OOPJ.java - Notepad++
coding Language Settings Tools Macro Run Plugins Window ?
DAC Mar24
Feb25 OOPJ.java
Test.java
1846
1847
-strictfp: ensure floating point
Iterable
extends
Collection
Map
e
e
e
List
Queue
Set
i
i
i
ArrayList
LinkedList
Vector
stack


Zoom Workplace Meeting View Edit Window Help
Tue 18 Mar 2:06 PM
098_Vaishnavi Kulkar...
035_Mahesh Ghule_JH
048_Kajal Palwe_KH
096_Rohini Bhondek...
Kiran Waghmare
124_Trushita Mahaja...
1846
1847
-strictfp: ensure floating point
Iterable
extends
Collection
Map
List
Queue
Set
ArrayList
LinkedList
Vector
Priority Queue
stack
Deque
ArrayDeque


Zoom Workplace Meeting View Edit Window Help
Tue 18 Mar 2:11 PM
098_Vaishnavi Kulkar...
048_Kajal Palwe_KH
085_Shubham Kumar_JH
108_Shital Kawthe_KH
Kiran Waghmare
092_Ravi Yadav_KH
-strictfp: ensure floating point
Iterable
Collection
List
Queue
Set
HashSet
LinkedHashSet
ArrayList
LinkedList
Vector
Priority Queue
stack
Deque
Sorted set
Treeset
ArrayDeque


Zoom Workplace Meeting View Edit Window Help
Tue 18 Mar 3:24 PM
us02web.zoom.us
098_Vaishnavi Kulkar...
Kiran Waghmare
088_puja Shingare_KH
043_Mrunal Shedge_...
078_Pooja Nichit_KH
Vipul Tembulwar
English
Hierarchy of Collection Framework in Java
Iterable
Collection
List
Queue
Set
ArrayList
LinkedList
Vector
Stack
Priority Queue
Dequeue
ArrayDequeue
HashSet
LinkedHashSet
SortedSet
TreeSet
Map
HashTable
LinkedHashMap
HashMap
SortedMap
TreeMap
Interface
Class
Implements
Extends

DAC Mar24 | Feb25 OOPJ.java | Test.java

```
1852  -readymade architecture composed of clasess and interface
1853
1854  ArrayList:
1855
1856  ArrayList a1 = new ArrayList();
1857  al.add(10);
1858  al.add("10");
1859  al.add(10);
1860
1861  ArrayList<String> a1 = new ArrayList<>();
1862  ArrayList a1 = new ArrayList();
1863
1864  al.add(10.0);
1865  al.add("10");//Error
1866  al.add(10.0);
1867
```

```
68
69  LinkedList l1 = new LinkedList();
70  LinkedList<Integer> l1 = new LinkedList<>();
71  l1.add(23);
72
73  Vector v = new Vector();
74  Vector<String> v = new Vector<>();
75  v.add("Abc");
76
77  Stack s1 = new Stack();
78  Stack<String> s1 = new Stack<>();
79  stack.push("Ketki");
```

DAC Mar24 | Feb25 OOPJ.java | Test.java

```
1881  stack.push("Ketki");
1882
1883  List Interface: (Ordered collection, Allow duplicates)
1884  Implements:
1885      -ArrayList : Fast retreival, slow insertion/deletion
1886      -LinkedList: Fast insertion/deletion
1887      -Vector : Legacy class
1888      -Stack  : LIFO(Last In First Out)
1889
1890      List l1 = new ArrayList();
1891      List l1 = new LinkedList();
```

```
    Collection l1 = new ArrayList();
    List<Integer> l1 = new ArrayList<>();

Set Interface:(Unique elements)
Implements:
    -HashSet : Unordered, fast access
    -LinkedHashSet : Maintains insertion order
    -TreeSet : Sorted order
```