

The screenshot shows a Notepad++ window with the title "Feb25 OOP.java". The code in the editor is as follows:

```
1553     - Object class
1554     - Wrapper class
1555     - Exception Handling
1556
1557 Object class:
1558 -----
1559
1560
1561
1562     -toString()
1563     -equals()
1564     -hashCode()
1565     -clone()
1566     -finalize()
1567     -wait()
1568     -notify()
1569     -notifyAll()
1570     -getClass()
```

The screenshot shows a Notepad++ window with the title "Feb25 OOP.java". The code in the editor is as follows:

```
1553     - Object class
1554     - Wrapper class
1555     - Exception Handling
1556
1557 Object class:
1558 -----
1559
1560
1561
1562     -toString() : Creating string representation
1563     -equals(Object obj) : comparision of objects
1564     -hashCode() : Generate hash code
1565     -clone() : Objects clone creation
1566     -finalize() : called by GC =>gc()
1567     -wait() : Thread method
1568     -notify() : Thread method
1569     -notifyAll() : Thread method
1570     -getClass() : Runtime class identification of object
```

```
View Encoding Language Settings Tools Macro Run Plugins Window 2
DAC Mar24 Feb25 OOP.java StringDemo.java ObjectDemo.java new 1 new 2
1553 - Object class
1554 - Wrapper class
1555 - Exception Handling
1556
1557
1558 Object class:
1559 -----
1560 Package: import java.lang.Object;
1561
1562 Every class is directly or indirectly derived from Object class
1563
1564
1565
1566 -toString() : Creating string representation
1567 -equals(Object obj) : comparision of objects
1568 -hashCode() : Generate hash code
1569 -clone() : Objects clone creation
1570 -finalize() : called by GC =>gc()
1571 -wait() : Thread method
1572 -notify() : Thread method
1573 -notifyAll() : Thread method
1574 -getClass() : Runtime class identification of object
```

The screenshot shows a video conference interface with five participants: Vaishnavi Kulkarni, Kiran Waghmare, Shubham Kumar, Ujjwal Patle, and Trushita Mahajan. Below the video feed, a browser window displays the Java API documentation for the `Object` class. The URL is `docs.oracle.com/javase/8/docs/api/java/lang/Object.html`. The page title is "Method Summary". The "All Methods" tab is selected. The table lists the following methods:

Modifier and Type	Method and Description
protected Object	<code>clone()</code> Creates and returns a copy of this object.
boolean	<code>equals(Object obj)</code> Indicates whether some other object is "equal to" this one.
protected void	<code>finalize()</code> Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
Class<?>	<code>getClass()</code> Returns the runtime class of this Object.
int	<code>hashCode()</code> Returns a hash code value for the object.
void	<code>notify()</code> Wakes up a single thread that is waiting on this object's monitor.
void	<code>notifyAll()</code> Wakes up all threads that are waiting on this object's monitor.

**Thank You Vaishnavi Anantrao Kulkarni.**

Congratulations! You have successfully booked a ticket.  
Booking details will be sent to Email/Mob: : va\*\*\*\*\*@gmail.com / 93\*\*\*\*\*60

**RAJYA RANI EXP (17612)** PNR: 8811349483

18:45 | C SHIVAJI MAH T 12:35 07:20 | H SAHIB NANDED  
Sun, 16 Mar Mon, 17 Mar

1 Adult | AC 3 Tier (3A) | Tatkal | Boarding at C Shivaji Mah T | Boarding Date: 16 Mar 2025 18:45  
Please check NTES website or NTES app for actual time before boarding

[View Cancellation Policy](#)

**Passenger Details**

1 VAISHNAVI ANANTR 24 yrs | Female | India | Middle | Aadhaar Verified  
Booking Status : CNF/B2/5/MB Current Status : CNF/B2/5/MB

[Book Return/Onward Ticket](#) [Book Another Ticket](#) [Book Connecting PNR](#) [Print Ticket](#)

**Introducing**

29°C Smoke

Gift Vouchers on activation and milestones\* HDFC BANK

098\_Vaishnavi Kulkarni\_JH Kiran Waghamare\_Vipul Tembulwar\_073\_Parag Palaskar\_KH 063\_Rahul Murmure\_JH 096\_Rohini Bhondekar\_KH

Object (Java Platform SE 8) docs.oracle.com/javase/8/docs/api/

**See Also:**  
Cloneable

**class Emp extends Object{**

**toString** @Override int toString();  
public String toString()

Returns:  
a string representation of the object. In general, the `toString` method returns a string that "textually represents" this object. The result should be a concise but informative representation that is easy for a person to read. It is recommended that all subclasses override this method.

The `toString` method for class `Object` returns a string consisting of the name of the class of which the object is an instance, the at-sign character `@', and the unsigned hexadecimal representation of the hash code of the object. In other words, this method returns a string equal to the value of:

```
getClass().getName() + '@' + Integer.toHexString(hashCode())
```

**Object**  
**Emp**

**notify**

public final void notify()

```

1562 -Every class is directly or indirectly derived from Object class
1563 -Methods provided by Object class:
1564     -toString() : Creating string representation
1565     -equals(Object obj) : comparison of Objects
1566     -hashCode() : Generate hash code
1567     -clone() : Objects clone creation
1568     -finalize() : called by GC =>gc()
1569     -wait() : Thread method
1570     -notify() : Thread method
1571     -notifyAll() : Thread method
1572     -getClass() : Runtime class identification of object
1573
1574 -toString() :
1575 -provides a string representation of an object
1576 -If not overridden, Exception: ClassNotFoundException#HexadecimalHashCode
1577
1578
1579
1580
1581
1582

```

```

graph TD
    Object --> Emp

```

```

String name;
int id;

HashCodeDemo(String name, int id) {
    this.name = name;
    this.id = id;
}

@Override
public int hashCode() {
    return id; //Using id as a unique hash code must be return
}

public static void main(String[] args) {
    HashCodeDemo t1 = new HashCodeDemo("Abc", 111);
    System.out.println(t1.hashCode());
}

```

**HashCode: (key-Value)**  

$$h(x): \text{key mod } x$$

Notepad++ window showing Java code for HashCodeDemo.java:

```
String name;
int id;

HashCodeDemo(String name, int id) {
    this.name = name;
    this.id = id;
}

@Override
public int hashCode() {
    return id; // Using id as a unique hash code must be return
}

public static void main(String[] args) {
    HashCodeDemo t1 = new HashCodeDemo("Abc", 111);
    System.out.println(t1.hashCode());
}
```

A hand-drawn diagram to the right illustrates a hash table mapping key values to their corresponding hash codes. The table has columns for key and value, with rows for 1, 2, 3, 4, and 5. A note above the table says "h(x):key mod x". Below the table, a calculation shows  $(20)_{10} \mod 4 = (4)_{10}$ .

Notepad++ window showing Java code for GetClassDemo.java:

```
class GetClassDemo {

    public static void main(String[] args) {
        GetClassDemo g1 = new GetClassDemo();
        System.out.println(g1.getClass().getName());
    }
}
```

Notes on the right side of the code explain Java's class hierarchy:

- Class ----> Object ? instanceof |true/false)
- Object -----> Class ? getClass()
- Object -----> Class (Name) ? getClass().getName()

Java source file

```
1 class GetClassDemo{  
2     public static void main(String[] args) {  
3         GetClassDemo g1 = new GetClassDemo();  
4         System.out.println(g1.getClass().getName());  
5     }  
6 }  
Note:  
-----  
Class -----> Object ? instanceof (true/false)  
Object -----> Class ? getClass()  
Object -----> Class (Name)? getClass().getName()
```

Java source file

```
1561 -Every class is directly or indirectly derived from Object class  
1562 -Methods provided by Object class:  
1563  
1564  
1565     -toString() : Creating string representation  
1566     -equals(Object obj) : comparision of objects  
1567     -hashCode() : Generate hash code  
1568  
1569     -clone() : Objects clone creation  
1570  
1571     -finalize() : called by GC =>gc()  
1572  
1573     -wait() : Thread method  
1574     -notify() : Thread method  
1575     -notifyAll() : Thread method  
1576  
1577     -getClass() : Runtime class identification of object  
1578  
1579     -toString() :  
1580     -provides a string representation of an object  
1581     -If not overriden, Exception: ClassNotFoundException#HexadecimalHashCode
```

