

Recording... | View

Kiran Waghamare

The screenshot shows a video conference interface with a participant named Kiran Waghamare. On the left, there is a code editor window displaying Java code for an exception demo. In the center, a terminal window shows the execution of the code and the resulting exception stack trace. A hand-drawn diagram of the Java exception hierarchy is overlaid on the terminal window, illustrating the inheritance path from Object to Throwable, then to Error and Exception, and finally to RunTime Exception.

```
class ExceptionDemo1{  
    public static void main(String[] args) {  
        int a=100;  
        int result = a/0;//throw s an exception  
        System.out.println(result);  
    }  
}  
try-cat C:\Test>
```

```
C:\Windows\system32 > + v  
C:\Test>javac ExceptionDemo1.java  
C:\Test>java ExceptionDemo1  
Exception in thread "main" java.lang.ArithmaticException: / by zero  
at ExceptionDemo1.main(ExceptionDemo1.java:6)  
C:\Test>
```

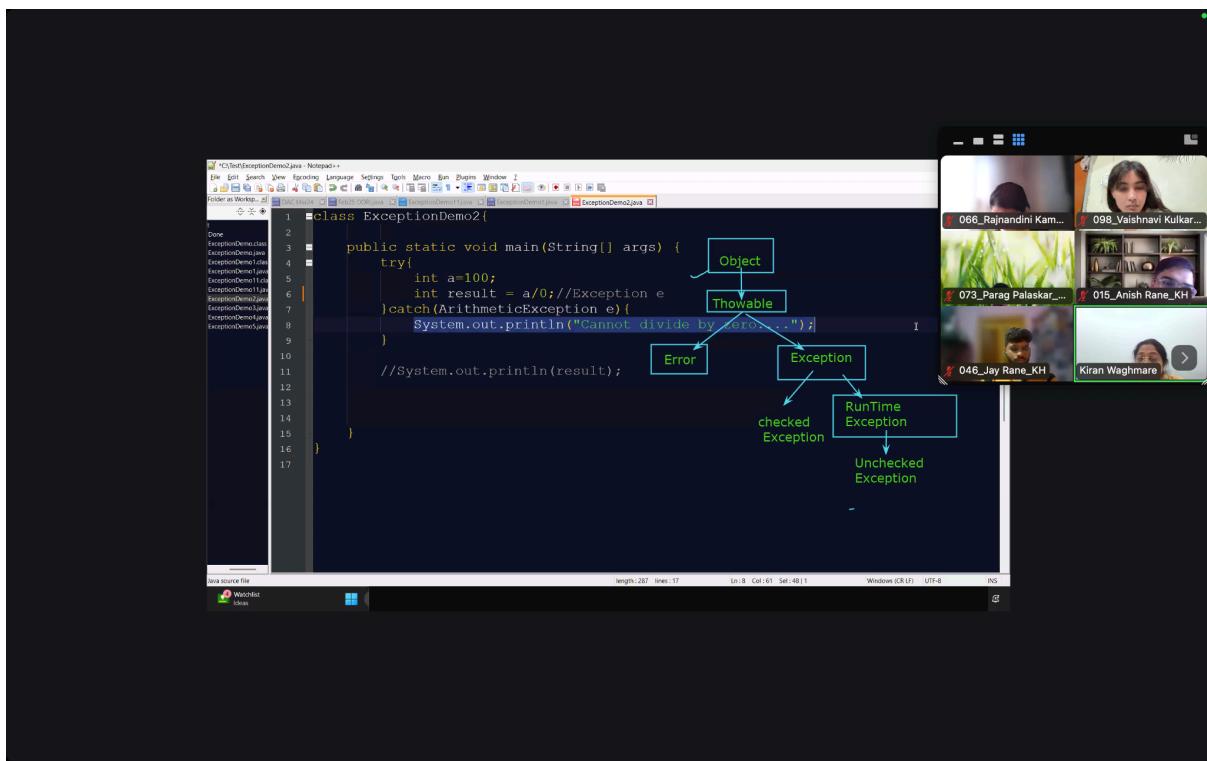
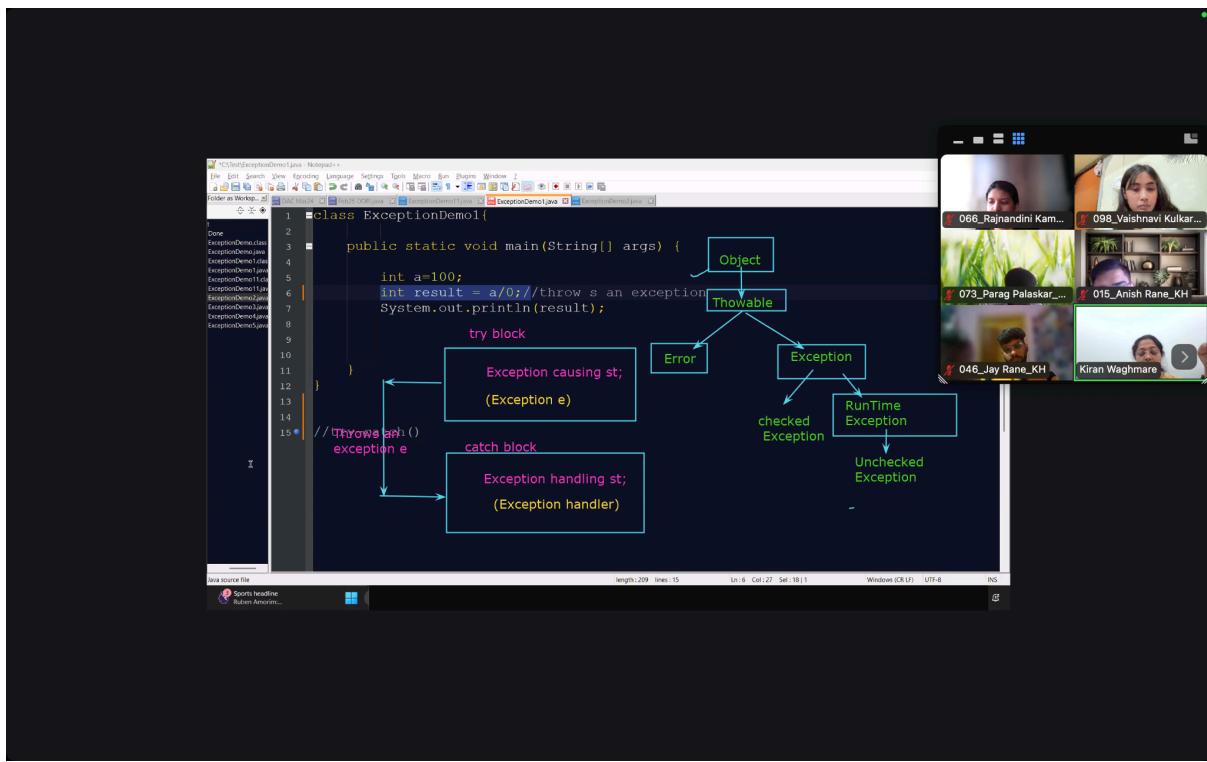
Object  
↓  
Throwable  
↓  
Error  
↓  
Exception  
↓  
RunTime Exception  
↓  
Unchecked Exception

The screenshot shows a video conference interface with a participant. On the left, there is a code editor window displaying Java code for an exception demo. In the center, a terminal window shows the execution of the code and the resulting exception stack trace. A hand-drawn diagram of the Java exception hierarchy is overlaid on the terminal window, illustrating the inheritance path from Object to Throwable, then to Error and Exception, and finally to RunTime Exception. The diagram also labels checked Exception as a subset of Exception.

```
class ExceptionDemo1{  
    public static void main(String[] args) {  
        int a=100;  
        int result = a/0;//throw s an exception  
        System.out.println(result);  
    }  
}  
try-cat C:\Test>
```

```
C:\Windows\system32 > + v  
C:\Test>javac ExceptionDemo1.java  
C:\Test>java ExceptionDemo1  
Exception in thread "main" java.lang.ArithmaticException: / by zero  
at ExceptionDemo1.main(ExceptionDemo1.java:6)  
C:\Test>
```

Object  
↓  
Throwable  
↓  
Error  
↓  
Exception  
↓  
checked Exception  
↓  
RunTime Exception  
↓  
Unchecked Exception



A screenshot of a Java IDE (NetBeans) showing a code editor with a Java file named `ExceptionDemo2.java`. The code demonstrates exception handling:

```
public static void main(String[] args) {
    System.out.println("Started");
    try{
        System.out.println("Entry point in try block");
        int a=10;
        int result = a/0;//Exception e
        System.out.println(result);
    }catch(ArithmeticException e){
        System.out.println("Cannot divide by zero.../user");
    }
    System.out.println("Finished");
}
```

To the right of the code editor is a diagram illustrating the Java exception hierarchy:

```
graph TD
    Object --> Throwable
    Throwable --> Error
    Throwable --> Exception
    Exception --> checkedException
    Exception --> RunTimeException
    RunTimeException --> uncheckedException
```

The Java exception hierarchy is shown as follows:

- `Object` is the root class.
- `Throwable` is a subclass of `Object`.
- `Error` and `Exception` are subclasses of `Throwable`.
- `Exception` is further divided into `checked Exception` and `RunTimeException`.
- `RunTimeException` is a subclass of `checked Exception`.
- `uncheckedException` is a subclass of `RunTimeException`.

In the bottom right corner of the screen, there is a video conference interface showing five participants in a grid:

- Row 1: 066\_Rajnandini Kam... and 098\_Vaishnavi Kulkar...
- Row 2: 073\_Parag Palaskar... and 046\_Jay Rane\_KH
- Row 3: Kiran Waghmare and 088\_puja Shingare\_KH

A screenshot of a terminal window titled "C:\Windows\system32" showing the compilation and execution of Java code. The terminal output is as follows:

```
C:\Test>javac ExceptionDemo1.java
C:\Test>java ExceptionDemo1
Exception in thread "main" java.lang.Arithmeti...
at exceptionDemo1.main(exceptionDemo1.java:6)
C:\Test>javac ExceptionDemo2.java
C:\Test>java ExceptionDemo2
Started
Entry point in try block
Cannot divide by zero...
Finished
C:\Test>
```

Annotations from the previous screenshot highlight the following lines of code in the `ExceptionDemo2.java` file:

- Line 10: `int result = a/0;//Exception e` (with a red circle around the division operator)
- Line 11: `System.out.println(result);`
- Line 12: `}catch(ArithmeticException e){`
- Line 13: `System.out.println("Cannot divide by zero.../user");`



A screenshot of a video conference interface with six participants in a 2x3 grid. The participants are:

- 066\_Rajnandini Kam... (top-left)
- 098\_Vaishnavi Kulkar... (top-right)
- 073\_Parag Palaskar... (middle-left)
- Kiran Waghmare (middle-right)
- 088\_puja Shingare\_KH (bottom-left)
- 080\_Projekta Dusane... (bottom-right)

The main window shows a Java code editor with the following code:

```
1 class A{}  
2 class B extends A{}  
3 class ExceptionDemo3{  
4  
5     public static void main(String[] args) {  
6         System.out.println("Started");  
7         A a = new A();  
8         B b = (B)a;//Downcasting: Exception e  
9         I  
10    System.out.println("Finished");  
11 }  
12 }  
13  
14  
15  
16 }  
17  
18 }
```

Information at the bottom of the screen includes: length: 269 lines: 18 Ln: 9 Col: 18 Sel: 9 | 1 Windows (XUL) UTF-8 ENG IN 11:03 AM 3/17/2025 IN



A screenshot of a video conference interface with six participants in a 2x3 grid. The participants are:

- 066\_Rajnandini Kam... (top-left)
- 098\_Vaishnavi Kulkar... (top-right)
- 073\_Parag Palaskar... (middle-left)
- Kiran Waghmare (middle-right)
- 088\_puja Shingare\_KH (bottom-left)
- 045\_Isha Srivastava... (bottom-right)

The main window shows a Java code editor with the following code. A terminal window is open to the right showing the output of running the code:

```
1 class A{}  
2 class B extends A{}  
3 class ExceptionDemo3{  
4  
5     public static void main(String[] args) {  
6         System.out.println("Started");  
7         A a = new A();  
8         try{  
9             B b = (B)a;//Downcasting: Exception e  
10        }catch(ClassCastException e){  
11            e.printStackTrace();  
12            System.out.println("Downcast me is issue hai!");  
13        }  
14  
15    }  
16  
17 }  
18  
19  
20  
21 }
```

The terminal output is:

```
C:\Test>java ExceptionDemo3  
Started  
java.lang.ClassCastException: A  
at ExceptionDemo3.main(  
C:\Test>  
Downcast me is issue hai!  
Finished  
C:\Test>
```

Information at the bottom of the screen includes: length: 392 lines: 23 Ln: 7 Col: 18 Row: 141 Windows (XUL) UTF-8 ENG IN 11:08 AM 3/17/2025 IN

