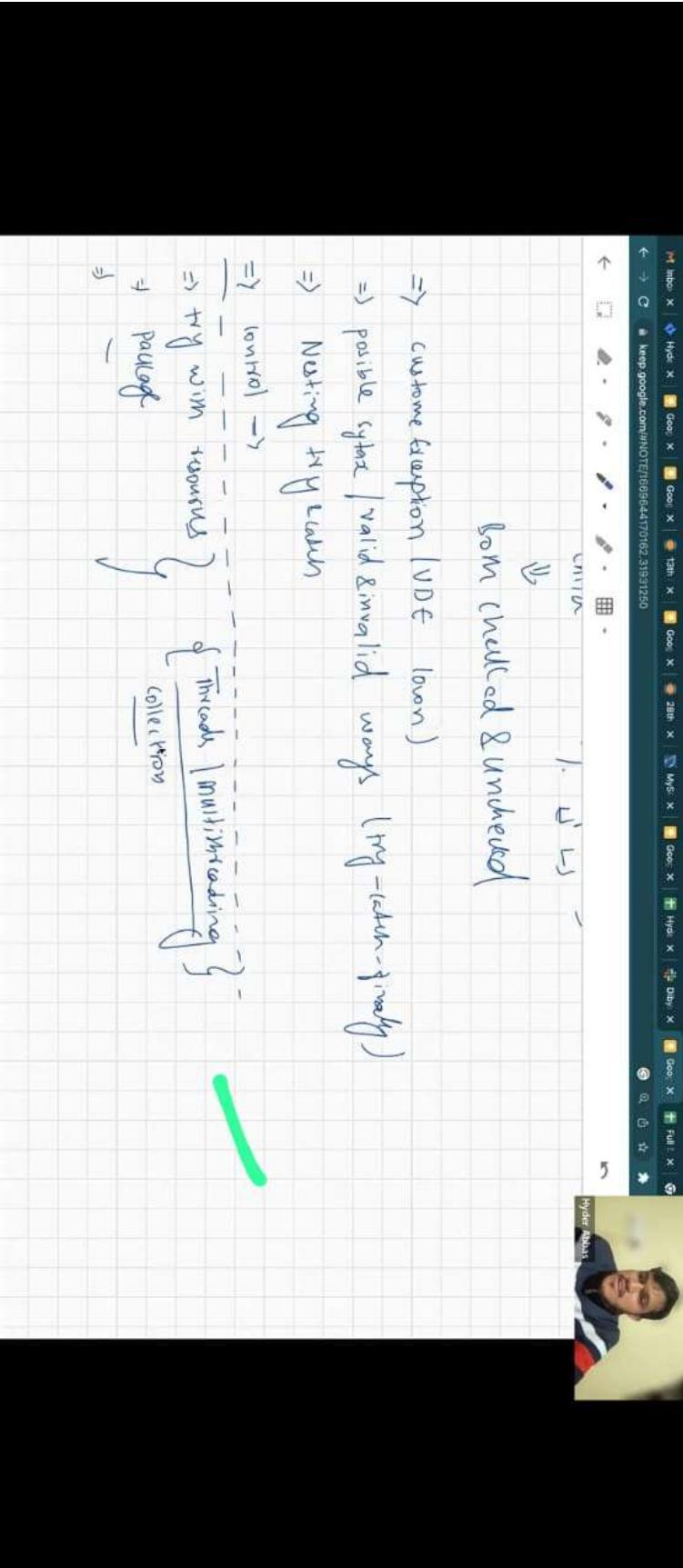


JAVA EXCEPTION Part3



JAVA

```
class Launch
{
    public static void main(String[] args)  JVM
    {
        try
        {
            System.out.println("Inside outer try");
            try
            {
                System.out.println("Inside inner try");
                System.out.println(10/0);
            }
            catch(ArithmeticException e)
            {
                System.out.println("Inside inner catch");
            }
            System.out.println("Outside inner try-catch");
        }
        catch(Exception e)
        {
            System.out.println("Inside outer catch");
        }
        finally
        {
            System.out.println("Inside outer finally");
        }
    }
}
```

OUTPUT

Inside outer try
Inside inner try
Inside inner catch
Outside inner try-catch
Inside outer finally

Hyder Abbas

© 2014 LIFETIME



```
class Launch
{
    public static void main(String[] args) JVM
    {
        try
        {
            System.out.println("Inside outer try");
            try
            {
                System.out.println("Inside inner try");
                System.out.println(10/0);
            } catch(NullPointerException e)
            {
                System.out.println("Inside inner catch");
            }
            System.out.println("Outside inner try-catch");
        } catch(Exception e)
        {
            System.out.println("Inside outer catch");
        } finally
        {
            System.out.println("Inside outer finally");
        }
    }
}
```

OUTPUT

Inside outer try
Inside inner try
Inside outer catch
Inside outer finally

Hyder Abbas



```
try {
    statement-1;
    statement-2;
    statement-3;
    try {
        statement-4;
        statement-5;
        statement-6;
    }
    catch(XXX e) {
        statement-7;
        finally {
            statement-8;
            statement-9;
        }
        catch(YYY e) {
            statement-10;
        }
        finally {
            statement-11;
        }
    statement-12;
}
```

4

CASE - 1 If no Exception occurs.

If an Exception occurs at statement-2 and the corresponding catch block is matched.

CASE - 2 If an Exception occurs at statement-2 and the corresponding catch block is not matched.

CASE - 3 If an Exception occurs at statement-2 and the corresponding catch block is not matched.



Hyder Abbas
© 2011 LIFELINE

```

try {
    statement-1;
    statement-2;
    statement-3;
    try {
        statement-4;
        statement-5; //
        statement-6;
    }
    catch(XXX e) {
        statement-7;
    }
    finally {
        statement-8;
        statement-9;
    }
    catch(YYY e) {
        statement-10;
    }
    finally {
        statement-11;
    }
    statement-12;
}

```

L

CASE-4
 If an Exception occurs at statement-5 and the corresponding inner catch block is matched.
 Statements - 1, 2, 3, 4, 7, 8, 9, 11 & 12 will be executed.
 Resulting in Normal Termination.

CASE-5
 If an Exception occurs at statement-5 and the corresponding inner catch block is not matched, but outer catch block is matched.
 Statements - 1, 2, 3, 4, 8, 10, 11 & 12 will be executed.
 Resulting in Normal Termination.

CASE-6
 If an Exception occurs at statement-5 and both inner and outer catch blocks are not matched.
 Statements - 1, 2, 3, 4, 8 & 11 will be executed.
 Resulting in Abnormal Termination.



```

try {
    statement-1;
    statement-2;
    statement-3;
    try {
        statement-4;
        statement-5;
        statement-6;
    } catch(XXX e) {
        statement-7; ██████████
    }
    finally {
        statement-8;
    }
    statement-9;
} catch(YYY e) {
    statement-10;
}
finally {
    statement-11;
}
statement-12;

```

6

CASE: 7 If an Exception occurs at Statement-7 and the corresponding catch block is matched.

If an Exception occurs at Statement-7 and the corresponding catch block is not matched.

CASE: 8 If an Exception occurs at Statement-7 and the corresponding catch block is not matched.

Statements - 1, 2, 3, (Exception may occur at one of the Statements - 4 or 5 or 6), 8, 10, 11, 12 will be Executed with Normal termination.

Statements - 1, 2, 3, (Exception may occur at one of the Statements - 4 or 5 or 6), 8, 11 will be Executed with Abnormal termination.



CASE - 1
Only try

CASE - 2
Only catch

CASE - 3
Only finally

CASE - 4
try - catch

CA
Reversing Order





CASE - 12

try-catch-finally

```
try
{
}
catch(XXX e)
{
}
finally
{}
```

CASE - 13

try - finally

```
try
{
}
finally
{}
```

CASE - 14

catch - finally

```
catch(XXX e)
try
{
}
finally
{}
```

CASE

Unordered

Hyder Abinav

```
try
{
}
finally
{
}
catch(XXX e)
```



CASE -16	try -multiple catch-finally
	<pre>try { try { } catch(XXX e) { } catch(YYY e) { } finally { } }</pre>
	<pre>try { try { } catch(XXX e) { } System.out.println("Hello") finally { } }</pre>
	<pre>try { try { } catch(XXX e) { } catch(YYY e) { } finally { } }</pre>



CASE -19

Only try with
resource

try (R)

10

CASE -24

Nested try-catch-finally

```
try
{
    try
    {
        catch(XXX e)
    }
    finally
    {
        }
    }
    catch(XXX e)
}
try
{
    catch(XXX e)
}
finally
{
    }
}
finally
{
    try
    {
        catch(XXX e)
    }
    finally
    {
        }
    }
}
```



Synchronous Exceptions

- Occurs at a specific program statement.

```
class Launch
{
    public static void main(String[] args)
    {
        String str = null;
        str.toUpperCase();
    }
}

class Launch
{
    public static void main(String[] args)
    {
        int[] a = new int[5];
        a[5] = 10;
    }
}
```



12



Asynchronous Exceptions

- Occurs anywhere in the program.

```
class Launch
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter your name:");
        String name = scan.nextLine();
        System.out.println("Enter your grades:");
        String grade = scan.nextLine();
    }
}
```

OUTPUT

```
Enter your name:  
Sachin  
Enter your grades:  
CTRL + C  
Keyboard Interrupt  
Exception in thread "main" Terminate batch job (Y/N)?
```

Examples:-
Keyboard interrupts,

Demonstration of Runtime Errors

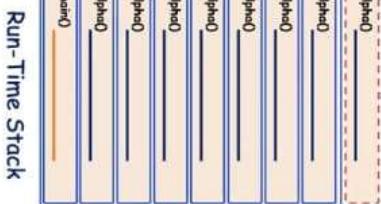
RECURSION

```
class Launch
{
    public static void alpha()
    {
        alpha();
    }

    public static void main(String[] args)
    {
        alpha();
    }
}
```

AR for
alpha()
AR for
alpha()

alpha0
alpha0
alpha0
alpha0
alpha0
alpha0
alpha0
alpha0



Run-Time Stack



6:37 AM

RECURSION

Demonstration of Runtime Errors

© 6.01 L1E.11 4.35



```
public static void alpha()
{
    beta();
}

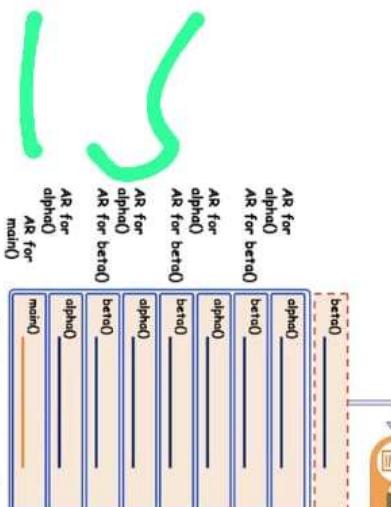
public static void beta()
{
    alpha();
}

public static void main(String[] args)
{
    alpha();
}
```

OUTPUT

```
Exception in thread "main" java.lang.StackOverflowError
at Launch.beta(Launch.java:9)
at Launch.alpha(Launch.java:5)
at Launch.beta(Launch.java:9)
at Launch.alpha(Launch.java:5)
...
```

Run-Time Stack





Demonstration of Runtime Errors

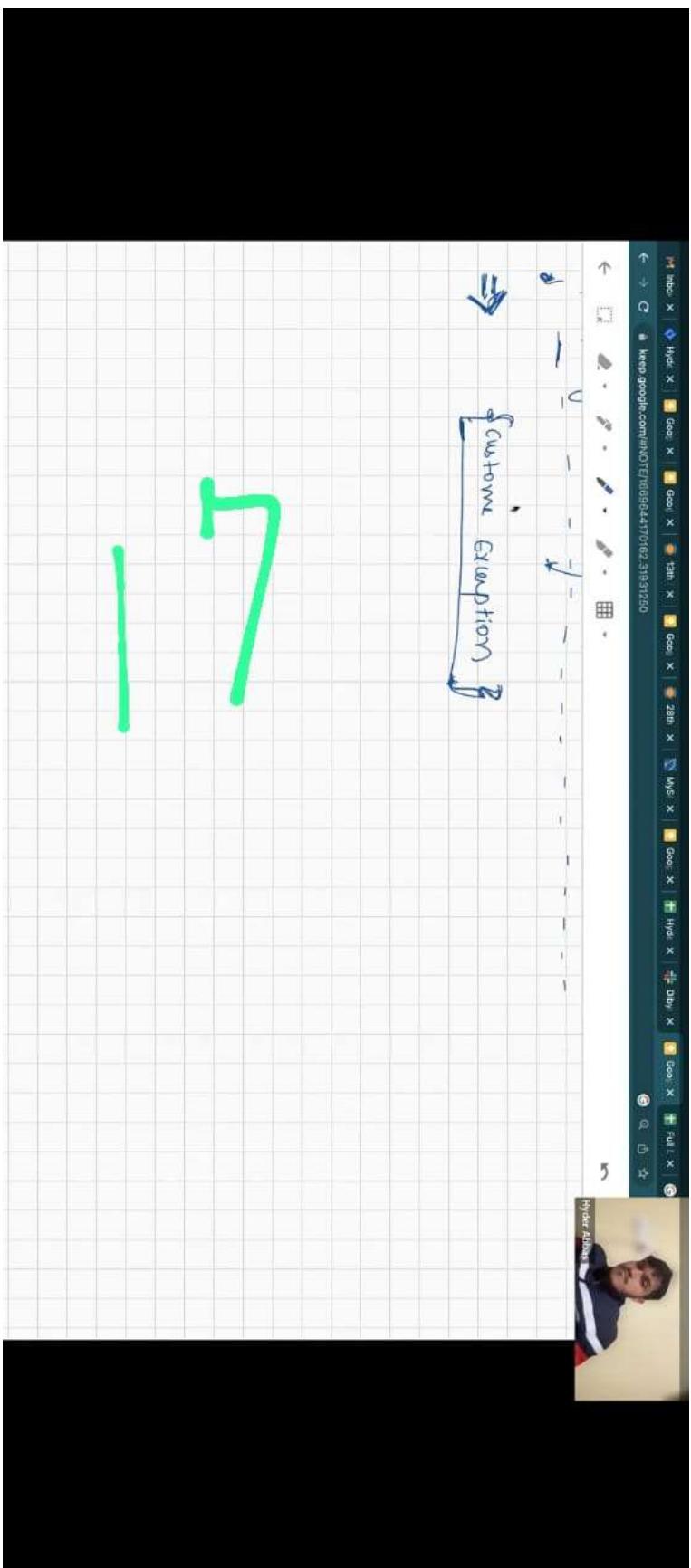
```
class Launch
{
    public static void main(String[] args)
    {
        int[] a = new int[1000000000];
    }
}
```

Impossible to **1000000000 * 4 = 4 GB** contiguously on the heap

OUTPUT

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
at Launch.main(Launch.java:5)
```

16



The screenshot shows a Java code editor within a development environment. The code is a Java program that prompts the user for a user ID and password, then checks if the user is a card owner.

```
1. package com;
2. import java.util.*;
3. 
4. public class ATM {
5.     Scanner scan=new Scanner(System.in);
6.     public void input() {
7.         System.out.println("Kindly enter the user id");
8.         uid=scan.nextInt();
9.         System.out.println("Kindly enter the pasword");
10.        pw=scan.nextInt();
11.    }
12.    public void verify() {
13.        System.out.println("Take your cash");
14.    }
15. }
16. 
17. class ATM {
18.     int uid;
19.     int pw;
20.     ATM() {
21.         uid=1234567890;
22.         pw=1234567890;
23.     }
24.     void withdraw() {
25.         System.out.println("Enter amount");
26.         int amount=scan.nextInt();
27.         if(amount>uid) {
28.             System.out.println("Insufficient balance");
29.         } else {
30.             uid-=amount;
31.             System.out.println("Your cash is "+uid);
32.         }
33.     }
34. }
```

Handwritten annotations are present in the code:

- A large green circle highlights the entire code block from line 1 to line 34.
- A vertical green line starts at the beginning of the code block and extends downwards, ending near the bottom of the code editor.



1 import java.util.Scanner;

2

3 class ATM

4 {

5 int userid=1212;

6 int password=1111;

7 int pw;

8 int uid;

9 public void input()

10 {

11 Scanner scan=new Scanner(System.in);

12 System.out.println("Kindly enter the user id");

13 uid=scan.nextInt();

14 }

15 System.out.println("Kindly enter the pasword");

16 pw=scan.nextInt();

17 System.out.println("Kindly enter the user id");

18 uid=scan.nextInt();

19 }

20 public void verify()

21 {

22 if(userid== uid) && (password == pw))

23 System.out.println("Take your cash");

24 }

25 else

26 {

27 System.out.println("Are you really card owner? bcz \n

Writble SmartSheet 47 / 22 / 2015

```
20
1  package com;
2  import java.util.*;
3  import java.io.*;
4  import java.awt.*;
5  import javax.swing.*;
6  import javax.swing.border.*;
7  import javax.swing.event.*;
8  import javax.swing.table.*;
9  import javax.swing.GroupLayout.*;
10 import javax.swing.LayoutStyle.*;
11 import javax.swing.GroupLayout.Alignment.*;
12 import javax.swing.LayoutStyle.ComponentPlacement.*;
13 import javax.swing.JButton.*;
14 import javax.swing.JFrame.*;
15 import javax.swing.JPanel.*;
16 import javax.swing.JScrollPane.*;
17 import javax.swing.JTable.*;
18 import javax.swing.JTable.*;
19 import javax.swing.JList.*;
20 import javax.swing.JList.*;
21 import javax.swing.JList.*;
22 import javax.swing.JList.*;
23 import javax.swing.JList.*;
24
25     else
26     {
27         System.out.println("Are you really card owner? ORZ");
28     }
29 }
30 }
31
32 class Bank
33 {
34     public void initiate()
35     {
36         Atm a=new Atm();
37         a.input();
38         a.verify();
39     }
40 }
41
42 public class LaunchCE {
43
44     public static void main(String[] args) {
45
46         Bank b=new Bank();
47         b.initiate();
48
49     }
50 }
```

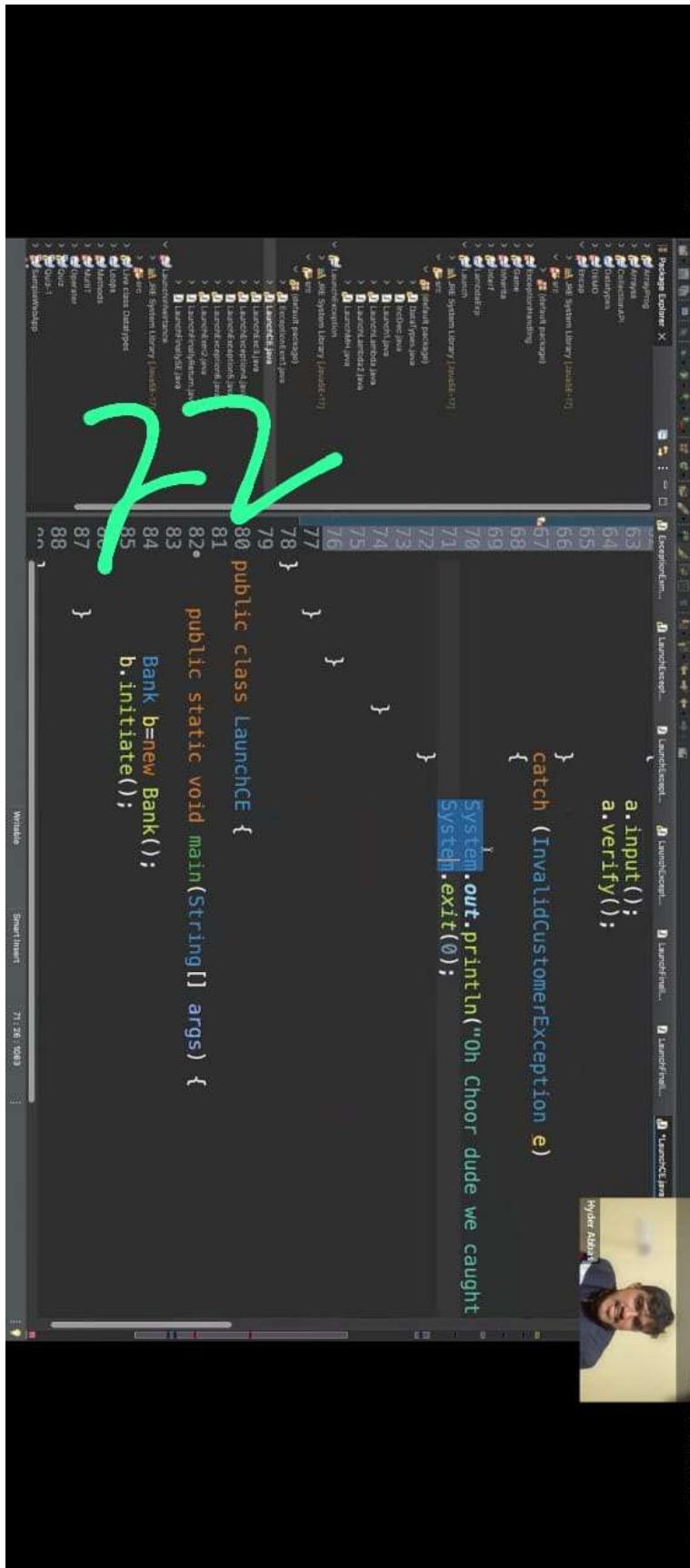


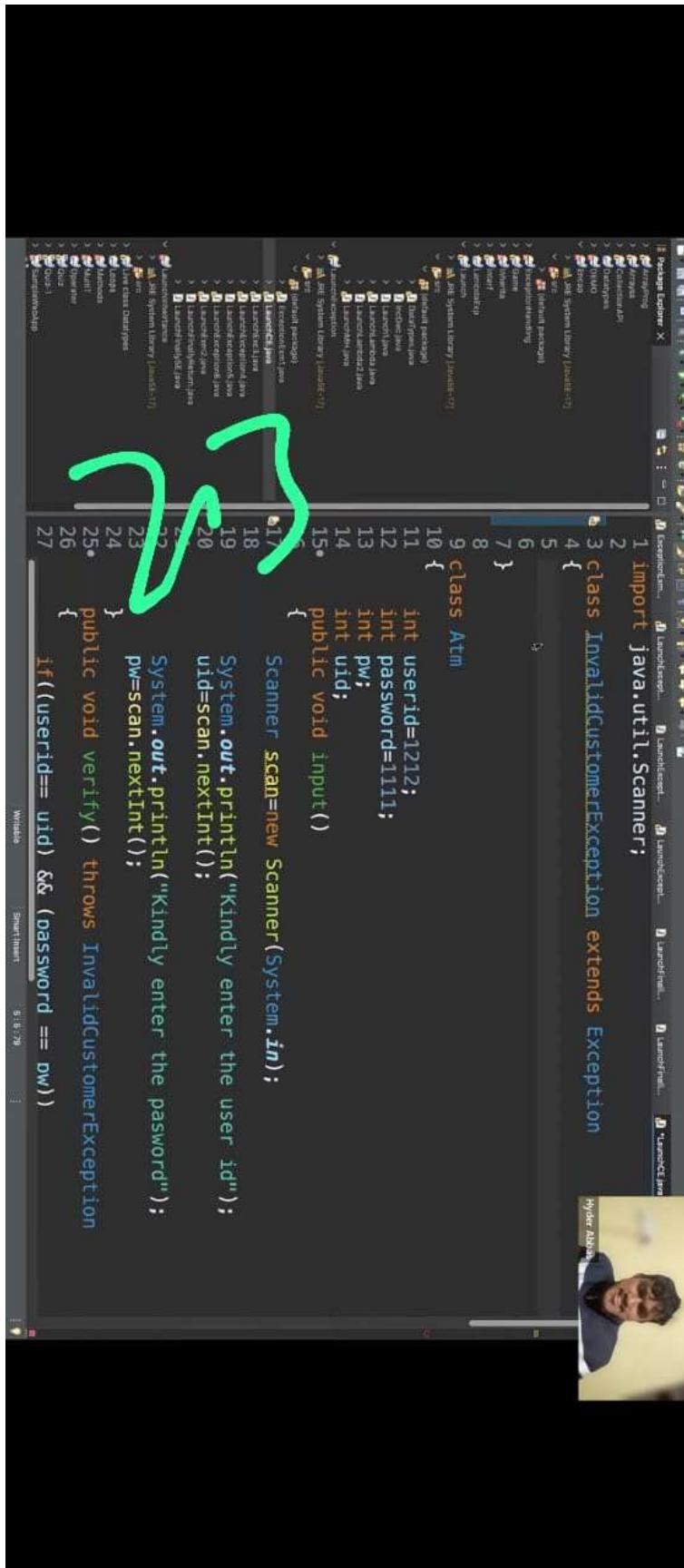
9

The screenshot shows a Java code editor with the following code:

```
1 package com.abc;
2
3 import java.util.Scanner;
4
5 public class Bank {
6     public void initiate() {
7         Scanner s=new Scanner(System.in);
8         System.out.println("Kindly enter the password");
9         String pw=s.nextLine();
10        if(pw.equals("1234")) {
11            System.out.println("Welcome to ABC Bank");
12            System.out.println("Enter your choice");
13            System.out.println("1. Withdrawal");
14            System.out.println("2. Deposit");
15            System.out.println("3. Balance Enquiry");
16            System.out.println("4. Change Password");
17            System.out.println("5. Exit");
18            int ch=s.nextInt();
19            switch(ch) {
20                case 1:
21                    withdrawal();
22                    break;
23                case 2:
24                    deposit();
25                    break;
26                case 3:
27                    balanceEnquiry();
28                    break;
29                case 4:
30                    changePassword();
31                    break;
32                case 5:
33                    System.out.println("Thank you for using ABC Bank");
34                    System.out.println("Have a nice day");
35                    System.out.println("Bye");
36                    System.out.println("-----");
37                    System.out.println("-----");
38            }
39        } else {
40            System.out.println("Incorrect Password");
41            System.out.println("Please try again");
42        }
43    }
44
45    public void withdrawal() {
46        System.out.println("Enter amount");
47        int amnt=s.nextInt();
48        if(amnt>0) {
49            System.out.println("Amount withdrawn is "+amnt);
50        } else {
51            System.out.println("Please enter valid amount");
52        }
53    }
54
55    public void deposit() {
56        System.out.println("Enter amount");
57        int amnt=s.nextInt();
58        if(amnt>0) {
59            System.out.println("Amount deposited is "+amnt);
60        } else {
61            System.out.println("Please enter valid amount");
62        }
63    }
64
65    public void balanceEnquiry() {
66        System.out.println("Enter account number");
67        int acno=s.nextInt();
68        if(acno>0) {
69            System.out.println("Balance available is "+acno);
70        } else {
71            System.out.println("Please enter valid account number");
72        }
73    }
74
75    public void changePassword() {
76        System.out.println("Enter old password");
77        String opw=s.nextLine();
78        System.out.println("Enter new password");
79        String npw=s.nextLine();
80        if(opw.equals("1234")) {
81            if(npw.equals("4321")) {
82                System.out.println("Password changed successfully");
83            } else {
84                System.out.println("New password must be different");
85            }
86        } else {
87            System.out.println("Old password is incorrect");
88        }
89    }
90 }
```

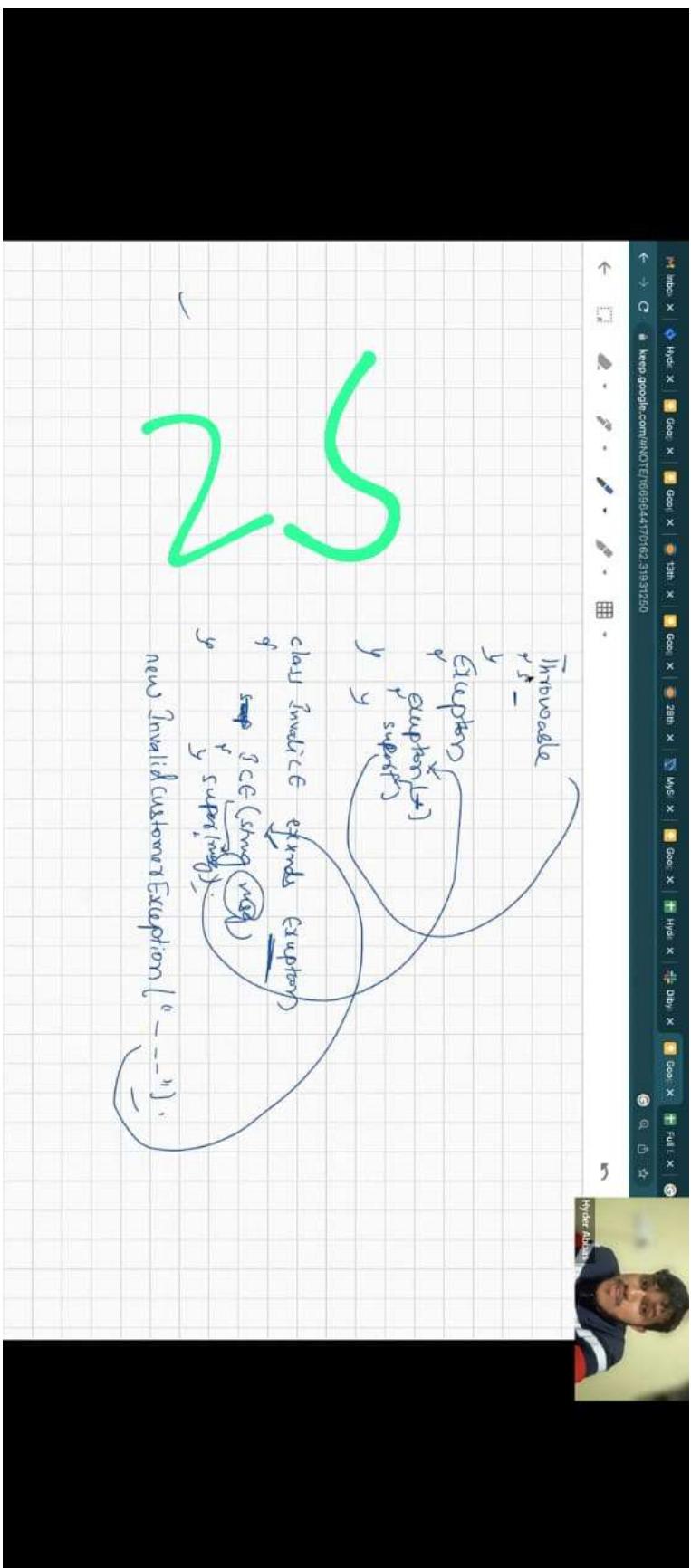
A large green checkmark is drawn over the code, covering lines 1 through 46. A blue arrow points from the checkmark down to the line "throw ice;".





The screenshot shows a Java code editor with a dark theme. A large green 'X' is drawn over the code area. The code defines a class named `InvalidCustomerException` that extends the `Exception` class. The code includes a constructor that takes a `String msg` parameter and calls the super constructor. It also includes a `Scanner` object to read user input for a user ID and password.

```
3 ass InvalidCustomerException extends Exception
4
5. public InvalidCustomerException(String msg)
6
7     super(msg);
8
9
10
11 ass Atm
12
13
14     int userId=1212;
15     int password=1111;
16     int pw;
17     int uid;
18
19     public void input()
20
21         Scanner scan=new Scanner(System.in);
22
23         System.out.println("Kindly enter the user id");
24         System.out.println("Kindly enter the pasword");
25         pw=scan.nextInt();
26
27     }
28
29     public void verify() throws InvalidCustomerException
```



↳ https://keep.google.com/p/NOTE/HB884470623991250

↳ class Throwable extends Object

↳

String msg;

public Throwable (String msg)

↳ thismsg = msg;

↳

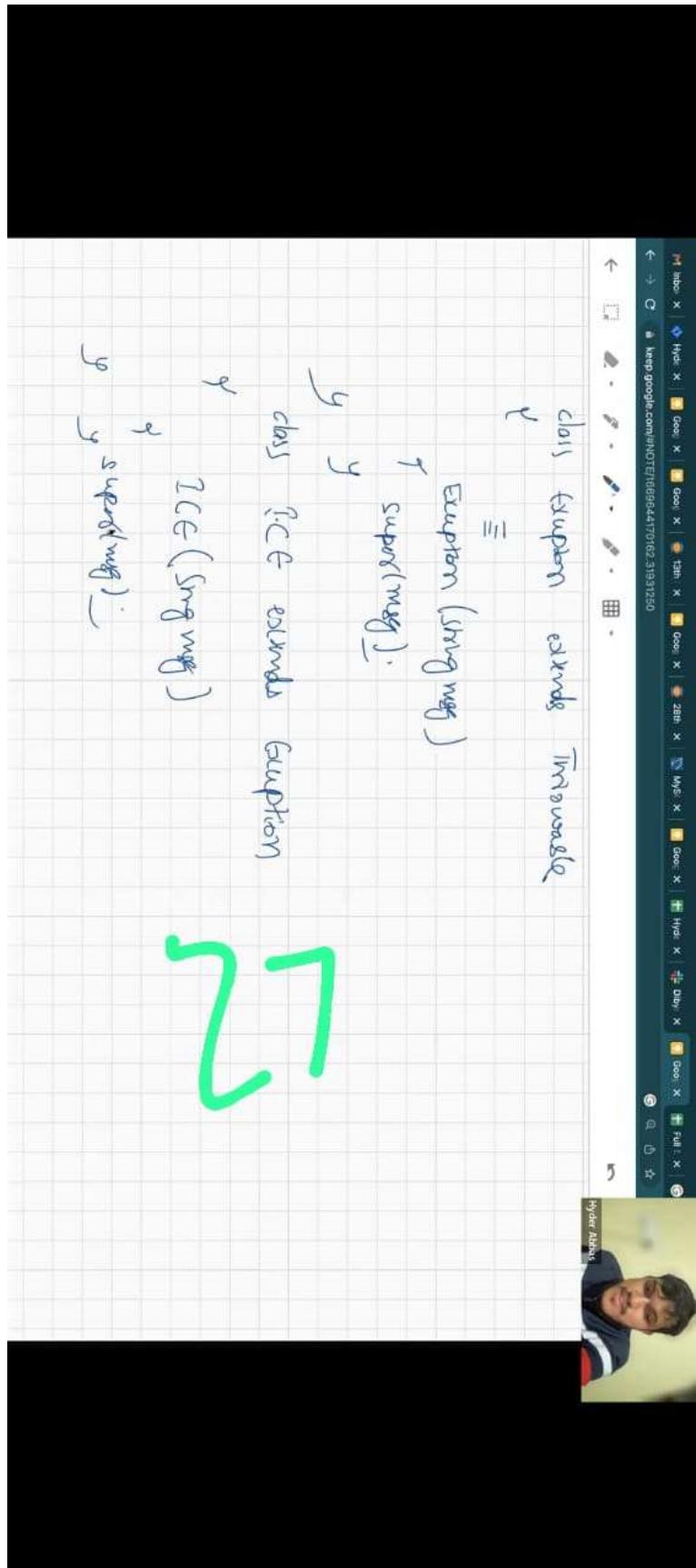
String getmsg ()

↳

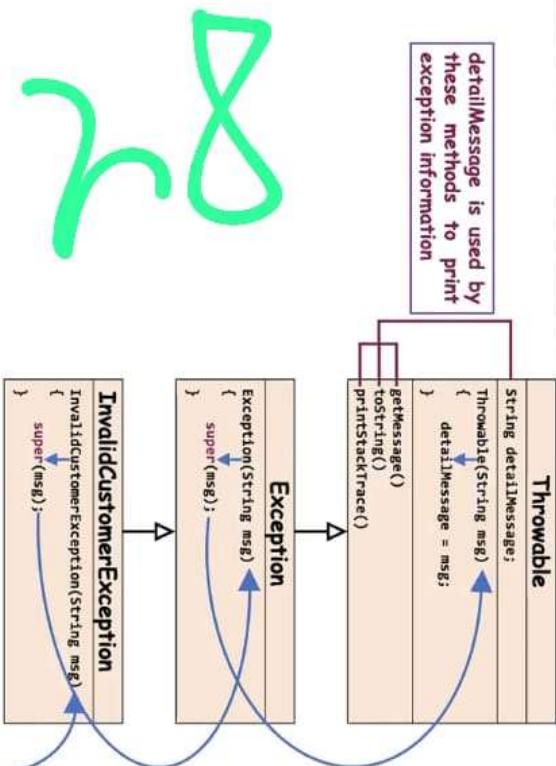
return msg;

↳

2



new InvalidCustomerException("Invalid input. Try again!");



detailMessage is used by
these methods to print
exception information



The screenshot shows a Java code editor with a dark theme. A large green 'C' is drawn over the code area, highlighting the class definition. The code is as follows:

```
1 import java.util.Scanner;
2 class UnderAgeException extends Exception
3 {
4     public UnderAgeException(String message)
5     {
6         super(message);
7     }
8 }
9 class OverAgeException extends Exception
10 {
11     public OverAgeException(String message)
12     {
13         super(message);
14     }
15 }
16 }
17
18
19 class Applicant
20 {
21     int age;
22     public void input()
23     {
24         Scanner scan=new Scanner(System.in);
25         System.out.println("Please enter your age");
26         age=scan.nextInt();
27     }
}
```

```
1/ 
18 |
19 class Applicant
20 {
21     int age;
22     public void input()
23     {
24         Scanner scan=new Scanner(System.in);
25         System.out.println("Please enter your age");
26         age=scan.nextInt();
27     }
28     void verify()
29     {
30         if(age <18)
31         {
32             System.out.println("You are not eligible to vote");
33         }
34         else if(age > 60)
35         {
36             System.out.println("You are eligible to vote");
37         }
38         else
39         {
40             System.out.println("You're eligible");
41         }
42     }
43 }
```

Writable SmartSheet 16:1:260

```
1  package Explorer; X
2  import java.awt.*;
3  import javax.swing.*;
4  import java.awt.event.*;
5  import java.awt.image.*;
6  import java.io.*;
7  import java.util.*;
8  import java.awt.print.*;
9  import java.awt.datatransfer.*;
10 import java.awt.Toolkit.*;
11 import java.awt.Container.*;
12 import java.awt.Window.*;
13 import java.awt.BorderLayout.*;
14 import java.awt.GridLayout.*;
15 import java.awt.GridBagLayout.*;
16 import java.awt.CardLayout.*;
17 import java.awt.FlowLayout.*;
18 import java.awt.List.*;
19 import java.awt.Menu.*;
20 import java.awt.MenuItem.*;
21 import java.awt.PopupMenu.*;
22 import java.awt.TextArea.*;
23 import java.awt.TextField.*;
24 import java.awt.TextArea.*;
25 import java.awt.TextArea.*;
26 import java.awt.TextArea.*;
27 import java.awt.TextArea.*;
28 import java.awt.TextArea.*;
29
30
31
32 w UnderAgeException("Ohh dude? calm down your time will come");
33 etMessage());
34
35
36
37
38 OverAgeException("your time is near calm down and pray");
39 etMessage());
40
41
42
43
44 re eligible");
45
46
47
48
49
50
51
52
53
54
55
```

The code is a Java application that handles user input for age. It checks if the user is under 18 and throws an exception if true. The exception message is "Ohh dude? calm down your time will come". The code then prints the message to the console.

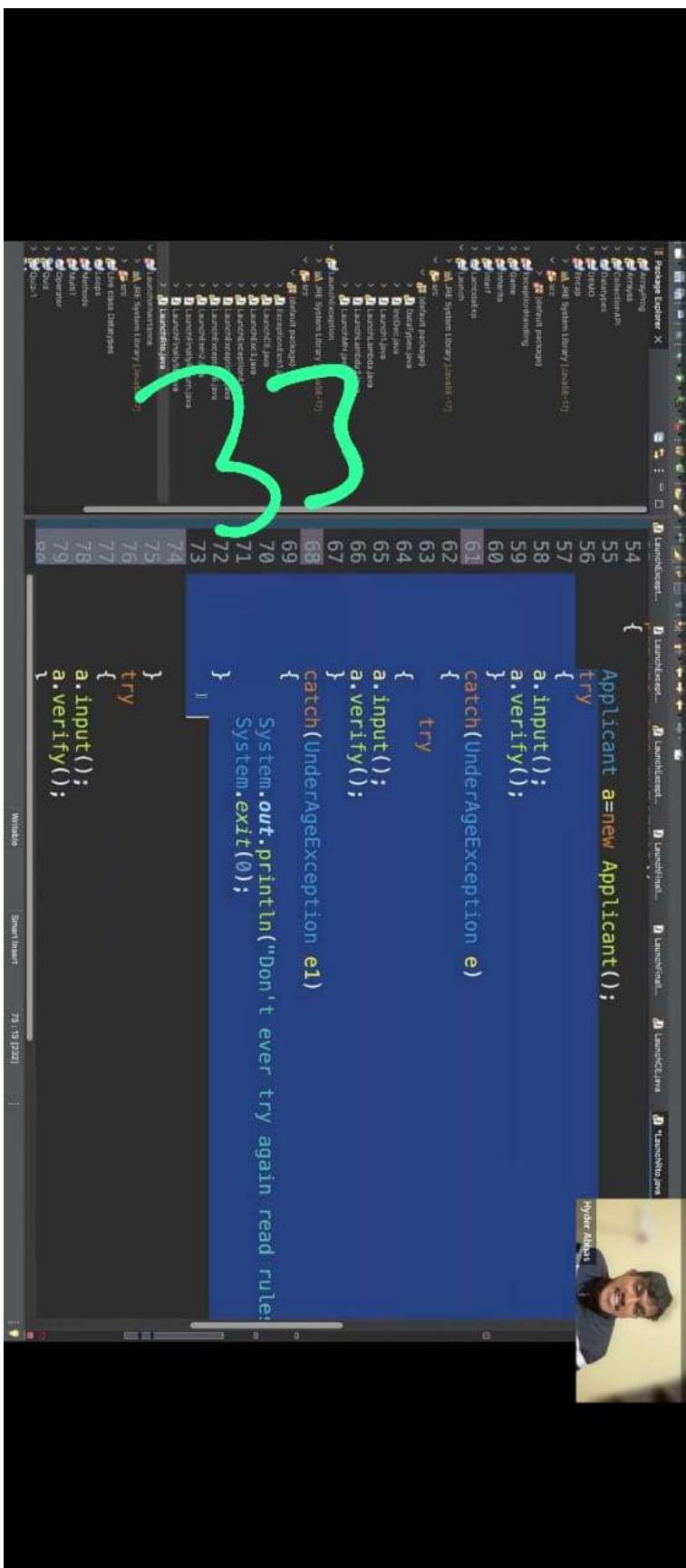
A green arrow points from the line "re eligible");" to the line "OverAgeException("your time is near calm down and pray");".





```
20 {
21     int age;
22     public void input()
23     {
24         Scanner scan=new Scanner(System.in);
25         System.out.println("Please enter your age");
26         age=scan.nextInt();
27     }
28     void verify() throws Exception
29     {
30         if(age < 18)
31             throw ue;
32     }
33     UnderAgeException ue=new UnderAgeException("Ohh du
34     are so young!");
35     ue.printStackTrace();
36     else if(age > 60)
37         throw oae;
38     OverAgeException oae=new OverAgeException("your time
39     is up!");
40     System.out.println(oae.getMessage());
41     throw oae;
42 }
43 else
44 {
45     System.out.println("You're eligible");
46 }
```





A screenshot of a Java IDE showing a code editor with the following Java code:

```
Applicant a=new Applicant();
try {
    a.input();
    a.verify();
} catch(UnderAgeException e) {
    try {
        a.input();
        a.verify();
    } catch(UnderAgeException e1) {
        System.out.println("Don't ever try again read rule:");
        System.exit(0);
    }
}
```

The code is annotated with a large green hand-drawn mark, which is a stylized number '3' with a vertical stroke through it, indicating a specific section or step in the code.

A screenshot of a Java development environment. The code editor shows a Java file named `LaunchRto.java` with the following content:

```
1 package Explorer;
2
3 import java.util.Scanner;
4
5 public class LaunchRto {
6     public static void main(String[] args) {
7         Scanner a = new Scanner(System.in);
8         System.out.println("Please enter your age: ");
9         int age = a.nextInt();
10        if (age < 18) {
11            System.out.println("Sorry, you are too young to drive! ");
12        } else if (age > 100) {
13            System.out.println("Sorry, that looks like a mistake! ");
14        } else {
15            System.out.println("You are old enough to drive! ");
16        }
17    }
18 }
```

The code editor has a dark theme with syntax highlighting. A large green 'X' is drawn over the top half of the code editor area. In the bottom right corner of the code editor, there is a small video player window showing a man's face.

The screenshot shows a Java code editor with the following code:

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();
        if (age < 18) {
            System.out.println("Sorry, you are too young to drive!");
        } else if (age > 100) {
            System.out.println("That's a very old age!");
        } else {
            System.out.println("You are " + age + " years old.");
        }
    }
}
```

Annotations and a green mark:

- Line 1: **1** **public class Main {**
- Line 2: **2** **public static void main(String[] args) {**
- Line 3: **3** **Scanner scanner = new Scanner(System.in);**
- Line 4: **4** **System.out.print("Enter your age: ");**
- Line 5: **5** **int age = scanner.nextInt();**
- Line 6: **6** **if (age < 18) {**
- Line 7: **7** **System.out.println("Sorry, you are too young to drive!");**
- Line 8: **8** **} else if (age > 100) {**
- Line 9: **9** **System.out.println("That's a very old age!");**
- Line 10: **10** **} else {**
- Line 11: **11** **System.out.println("You are " + age + " years old.");**
- Line 12: **12** **}**
- Line 13: **13** **}**
- Line 14: **14** **}**

A large green hand-drawn mark is drawn over the lines from 6 to 12.

The screenshot shows a Java development environment with a code editor and a video player window.

Code Editor:

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Please enter your age: ");
        int age = scanner.nextInt();
        if (age < 18) {
            System.out.println("Oh dude? calm down your time will come");
        } else {
            System.out.println("Please enter your age: ");
            int age2 = scanner.nextInt();
            if (age2 < 18) {
                System.out.println("Oh dude? calm down your time will come");
            } else {
                System.out.println("Don't ever try again read rules");
            }
        }
    }
}
```

A green wavy underline highlights the word "else" in the first conditional block.

Video Player:

A video player window titled "By der Abbas" is visible in the background. The video thumbnail shows a man with a beard. The video progress bar indicates it is at 1:11 of 1:11. The video title is "Java exception handling tutorial - Java exception handling tutorial".

```

class Thingy { meter=m = new Meter(); }

class Component { void go() { System.out.print("c"); } }

class Meter extends Component {
    void go() { System.out.print("m"); } // overriding method
}

class DeluxeThingy extends Thingy {
    public static void main(String[] args) {
        DeluxeThingy dt = new DeluxeThingy();
        dt.m.go();
        Thingy t = new DeluxeThingy();
        t.m.go();
    }
}

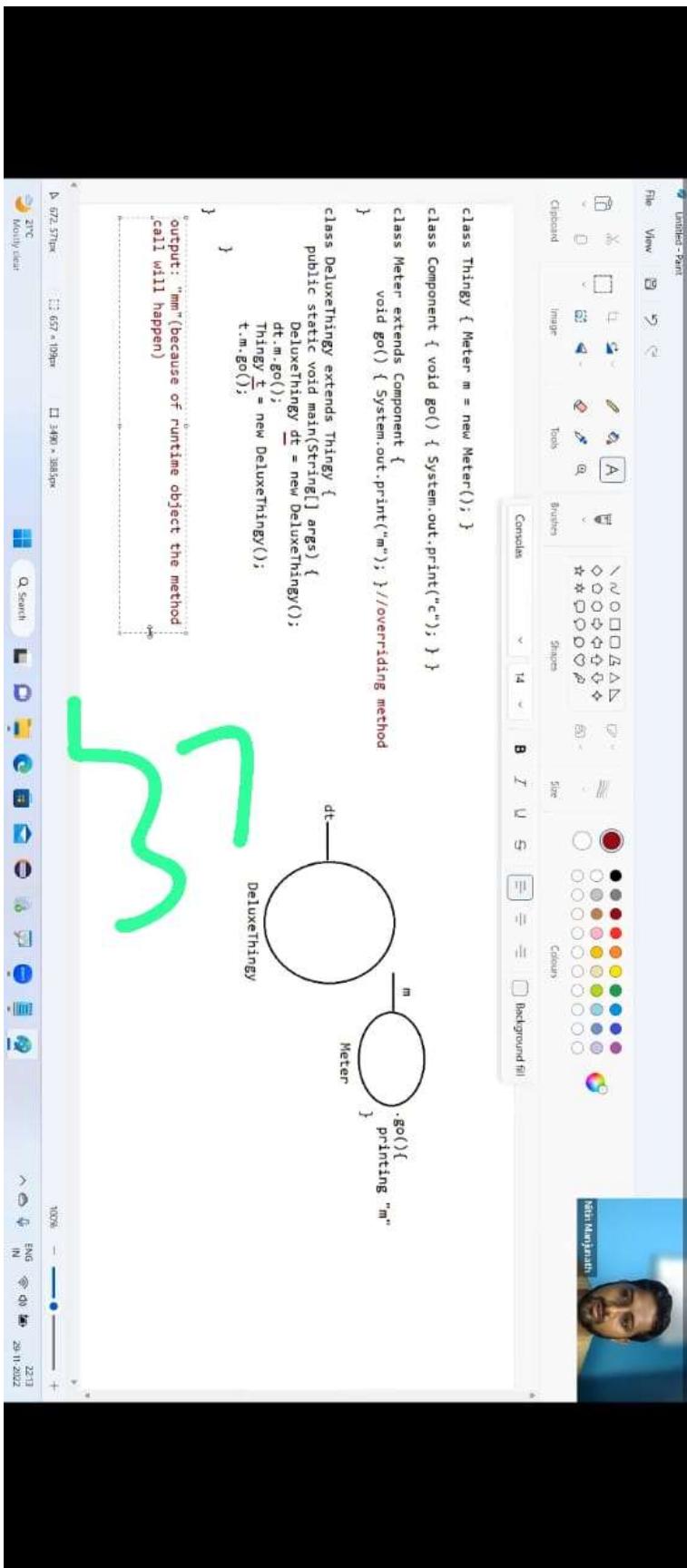
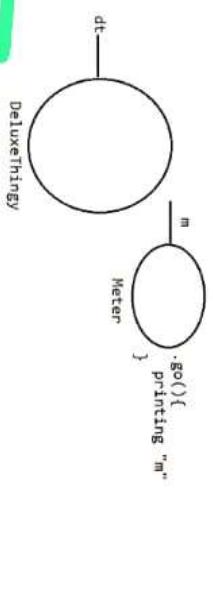
} // end class DeluxeThingy

output: "mm" (because of runtime object the method
call will happen)

```

dt → m

m .go() printing "m"



```
29.11.2022 /nijpenni-classnotes - Notepad  
File Edit View  
class Thingy { Meter m = new Meter(); }  
class Component void go() { System.out.print("c"); } }  
class Meter extends Component void go() { System.out.print("m"); } }  
class DeluxeThingy extends Thingy {  
    //meter object would come and meter would extend component so navigation is possible  
    public static void main(String[] args) {  
        DeluxeThingy dt = new DeluxeThingy();  
        dt.m.go();  
        Thingy t = new DeluxeThingy();  
        t.m.go();  
    }  
}  
Which two are true? (Choose two.)  
A. The output is mm.  
B. The output is mc.  
C. Component is-a Meter.  
D. Component has-a Meter.  
E. DeluxeThingy is-a Component.  
F. DeluxeThingy has-a Component.  
  
relationship b/w 2 classes as inheritance => IS-A relation  
relationship b/w 2 classes which are used for navigation => HAS-A relationship
```

29.11.2022 /nijpenni-classnotes - Notepad

File Edit View

class Thingy { Meter m = new Meter(); }
class Component void go() { System.out.print("c"); } }
class Meter extends Component void go() { System.out.print("m"); } }
class DeluxeThingy extends Thingy {
 //meter object would come and meter would extend component so navigation is possible
 public static void main(String[] args) {
 DeluxeThingy dt = new DeluxeThingy();
 dt.m.go();
 Thingy t = new DeluxeThingy();
 t.m.go();
 }
}

Which two are true? (Choose two.)

A. The output is mm.
B. The output is mc.
C. Component is-a Meter.
D. Component has-a Meter.
E. DeluxeThingy is-a Component.
F. DeluxeThingy has-a Component.

relationship b/w 2 classes as inheritance => IS-A relation
relationship b/w 2 classes which are used for navigation => HAS-A relationship



```
File Edit View  
29.11.2022 JavaPlec_Classes01 - Notepad  
  
//meter object would come and meter would extend component so navigation is possible  
public static void main(String[] args) {  
    DeluxeThingy dt = new DeluxeThingy();  
    dt.m.go();  
    Thingy t = new DeluxeThingy();  
    t.m.go();  
}  
  
Which two are true? (Choose two.)  
A. The output is mm.  
B. The output is mc.  
C. Component is-a Meter.  
D. Component has-a Meter.  
E. DeluxeThingy is-a Component.  
F. DeluxeThingy has-a Component.  
  
relationship b/w 2 classes as inheritance => IS-A relation  
relationship b/w 2 classes which are used for navigation => HAS-A relationship  
  
Answer: A,F
```

37

```
29.11.2022 snippets_classmate - Notepad  
File Edit View  
Q>  
public class Test{  
    public static void main(String[] args){  
        boolean a = new Boolean(Boolean.valueOf(args[0]));  
        System.out.println(a);  
    }  
}  
And the commands are  
javac Test.java  
java Test 1 null
```

What is the result?

- A. 1 null
- B. true false
- C. false false
- D. true true
- E. Class cast Exception is thrown at the runtime

40

21°C
mostly clear



10:00 | Windows (CEP) | UTF-8
^ ⌂ ⌂ ENG ⌂ ⌂ 29.11.22:19
N



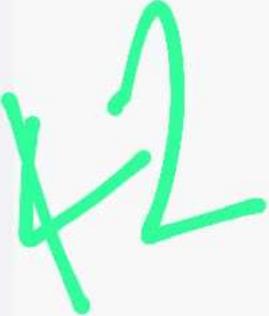
```
4.41  HW 19/10/2022, 09:59:00 - Notepad  
File Edit View  
System.out.println(b); //false  
}  
And the commands are  
javac Test.java  
java Test 1 null  
  
commandline arguments  
=====  
args[0] = "1"  
args[1] = "null"  
  
What is the result?  
A. 1 null  
B. true false  
C. false false  
D. true true  
E. Class cast Exception is thrown at the runtime
```

Answer: C

41

```
29.11.2022 /imports_classes01 - Notepad  
File Edit View  
Q>  
public class Test {  
    public static void doSum(int x, int y){  
        System.out.println("int sum is:: "+(x+y));  
    }  
    public static void doSum(Integer x, Integer y){  
        System.out.println("integer sum is:: "+(x+y));  
    }  
    public static void doSum(double x, double y){  
        System.out.println("double sum is:: "+(x+y));  
    }  
    public static void doSum(float x, float y){  
        System.out.println("float sum is:: "+(x+y));  
    }  
    public static void main(String[] args) {  
        doSum(10,20);  
        doSum(10.0,20.0);  
    }  
}
```

What is the result?
A. int sum is :: 30



Unacademy

```
29.11.2022 Unacademy - Numbers
File Edit View
public static void main(String[] args) {
    doSum(10.20);
    doSum(10.0.20.0);
}
```

What is the result?

- A. int sum is :: 30
float sum is :: 30.0
- B. int sum is :: 30
double sum is :: 30.0
- C. Integer sum is :: 30
double sum is :: 30.0
- D. Integer sum is:: 30
float sum is :: 30.0

Answer: B

43

Unacademy



Unacademy



Unacademy



File Edit View

Given:

```
1. public class BuildStuff{  
2.     public static void main(String[] args){  
3.         Boolean test = new Boolean(true); //test = true  
4.         Integer x = 343;  
5.         Integer y = new BuildStuff().go(test, x); // true,343  
6.         System.out.println(y); //49  
7.     }  
8.     int go(Boolean b, int i){  
9.         if(b) return (i/7); // return 343/7 => 49  
10.    return (i/49);  
11. }  
12.}
```

What is the result?

A. 7
B. 49
C. 343
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: B

44

29.11.2022 20:59:56 - NutriRock

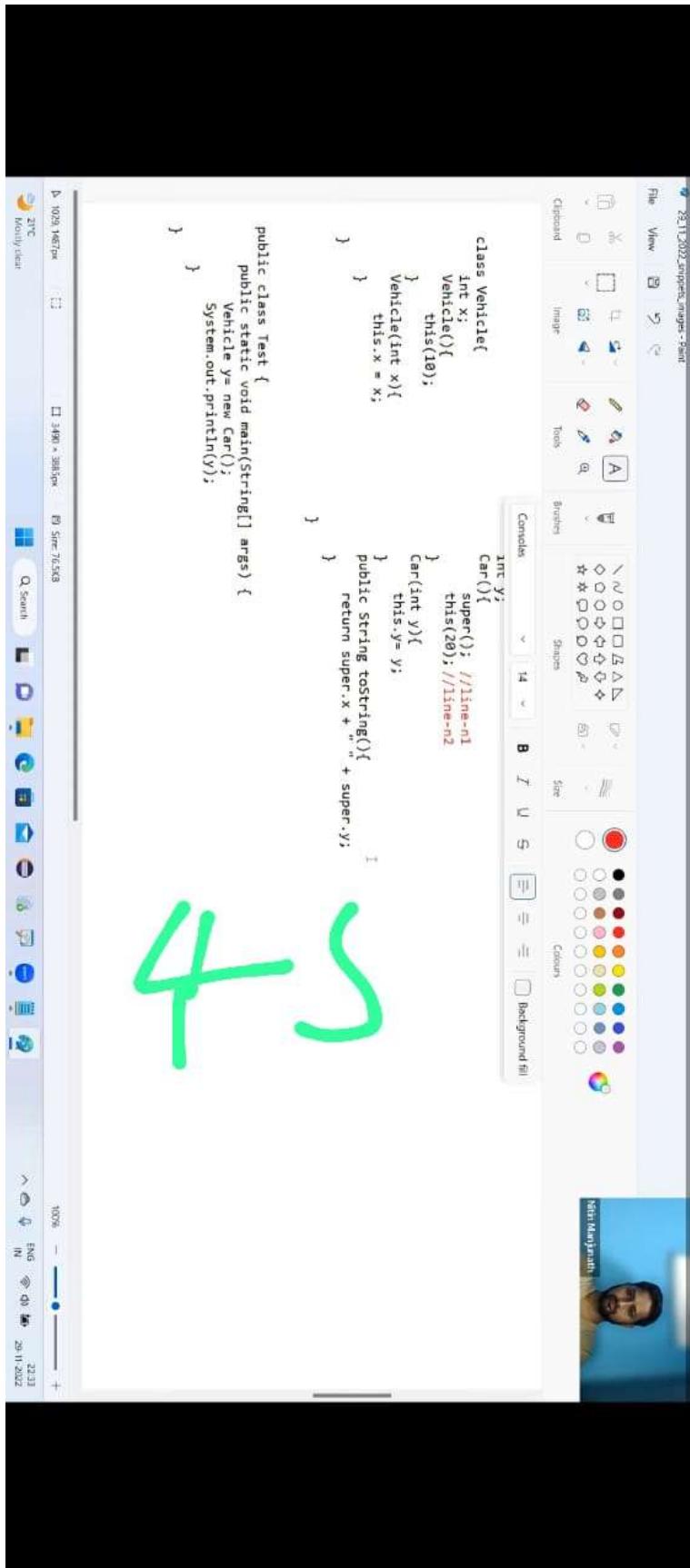
UniTutor

21°C
mostly clear



100% Windows (CEP) | UTF-8

ENG ⌂ ENG 22:29
29.11.2022



29.11.2022/Untitled - Notepad

```
File Edit View  
this(20); //line n2  
}  
Car(int y){  
    this.y=y;  
}  
public String toString(){  
    return super.x + " " + super.y;  
}  
}  
public class Test{  
    public static void main(String[] args){  
        Vehicle v=new Car();  
        System.out.println(v);  
    }  
}
```

Predict the answer

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1
- D. Compilation fails at line n2

Answer: D

46

Un 180°C/171°F 99% Windows (CEP) 177°/8 ENG ⌂ 29.11.2022

21°C Mostly clear

Q Search

File Edit View

- C. Compilation fails at line n1
D. Compilation fails at line n2

Answer: D

```
public static void main(String[] args) {  
    Integer i = new Integer(1) + new Integer(2); //line-12  
    switch(i) //line-13  
    {  
        case 3: System.out.println("three"); break;  
        default: System.out.println("other"); break; //line-15  
    }  
}
```

What is the result?

- C. An exception is thrown at runtime.
 - D. Compilation fails because of an error on line 12.
 - E. Compilation fails because of an error on line 13.
 - F. Compilation fails because of an error on line 15.

L'ÎLE COLT

47

29.11.2022/inputs_classmate - Notepad

File Edit View

Answer: A



Q>
Given:
public class Foo {
 static int[] a;
 static { a[0]=2; }
 public static void main(String[] args) {
 }
}
Which exception or error will be thrown when a programmer attempts to run this code?
A. java.lang.StackOverflowError
B. java.lang.IllegalStateException
C. java.lang.ExceptionInInitializerError
D. java.lang.ArrayIndexOutOfBoundsException

48

static{
//if some exception occurs during initialization then it is called as "ExceptionInInitializerError"
}

Un 172.60.18 99% Windows (CELF) 22.99
21°C Mon 11/29 2022
Q. Search

29.11.2022 JavaPenc_Classes - NutriPack

File Edit View

Q>

Given:

```
10. interface Jumper { public void jump(); }
```

...

```
20. class Animal {}
```

...

```
30. class Dog extends Animal { //Dog Is-A Animal, Dog Has-A tail
```

31.

```
    Tail tail;
```

32. }

...

```
40. class Beagle extends Dog implements Jumper { //Beagle is a Animal, beagle has-a tail, Beagle is-A jumper
```

41.

```
    public void jump() {}
```

42. }

...

```
50. class Cat implements Jumper {
```

51.

```
    public void jump() {}
```

52. }

Which three are true? (Choose three.)

A. Cat Is-a Animal
B. Cat Is-a Jumper
C. Dog Is-a Animal
D. Dog Is-a Jumper
E. Cat Has-a Animal

Unlocked 50%
21°C
mostly clear

Q Search

Windows (CELF)
UTF-8

ENG ☰ 29.11.2022 22:44

49

The screenshot shows a Microsoft Word document with the following content:

```
File Edit View  
29.11.2022 - subject-classnotes - Notepad  
30. class Dog extends Animal {  
    Tail tail;  
}  
31.  
32. }  
...  
40. class Beagle extends Dog implements Jumper {  
    public void jump() {}  
41.  
42. }  
...  
50. class Cat implements Jumper {  
    public void jump() {}  
51.  
52. }  
Which three are true? (Choose three.)  
A. Cat is-a Animal  
B. Cat is-a Jumper  
C. Dog is-a Animal  
D. Dog is-a Jumper  
E. Cat has-a Animal  
F. Beagle has-a Tail  
G. Beagle has-a Jumper
```

Handwritten note: SO

Answer: BCF

```
1. class X{  
2.     X() {System.out.print(1);}  
3.     X(int x){  
4.         this(); System.out.print(2);  
5.     }  
6. }  
7. public class Y extends X{  
8.     Y() {super(6); System.out.print(3);}  
9.     Y(int y){  
10.        this(); System.out.println(4);  
11.    }  
12.    public static void main(String[] a) { new Y(5); }  
13. }
```

What is the result?
A. 17

- C. 1234
D. 2134
E. 2143
F. 4321

Answer: C(1234)

Un 237, Col 16

51