

# JAVA EXCEPTION Part3

↳ born checked & unchecked

⇒ custom exception (VDE loan)

⇒ possible syntax / valid & invalid ways (try-catch-finally)

⇒ Nesting try catch

⇒ control →

⇒ try with resources

⇒ package

⇒

{ threads / multithreading }  
collection



```

class Launch
{
    public static void main(String[] args)
    {
        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
    
```



```

class Launch
{
    public static void main(String[] args)
    {
        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

```





```

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.

Statements - 1, 2, 3, 4, 5, 6, 8, 9, 11 & 12 will be executed.  
Resulting in Normal Termination.

**CASE # 2**

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

Statements - 1, 10, 11 & 12 will be executed.  
Resulting in Normal Termination.

**CASE # 3**

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

Statements - 1 & 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;

```

15

#### 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;

```

#### CASE # 7

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

#### 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, 11 will be Executed with Abnormal termination.

9



Syed Abbas

CASE - 1

Only try

```
try
{
}
```



CASE - 2

Only catch

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



CASE - 3

Only finally

```
finally
{
}
```



CASE - 4

try - catch

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



CASE - 5

Reverse Order

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



CASE - 6

Multiple try

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



CASE - 7

Multiple try

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



CASE - 8

Multiple try - catch

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



CASE - 9

Multiple catch

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



CASE - 10

Multiple catch

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



CASE - 11

Multiple catch

```
try
{
}
catch(XXX | VVV e)
{
}
```



Multithreaded Access





CASE - 12

try-catch-finally

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



CASE - 13

try - finally

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



CASE - 14

catch - finally

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



CASE - 15

Unordered

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





CASE - 16

try-multiple  
catch-finally

```
try
{
}
catch (xxx e)
{
}
catch (yyy e)
{
}
finally
{
}
```



CASE - 17

Multiple  
finally

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



CASE - 18

Statements in-between  
try-catch-finally

```
try
{
}
{
System.out.println("Hi");
catch (xxx e)
{
}
}
System.out.println("Hello")
;
finally
{
}
```



CASE - 19

Only try with  
resource

```
try (R)
{
}
}
```

10

# CASE - 24

## Nested try-catch-finally

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

OUTPUT

java.lang.NullPointerException  
at Launch.main(Launch.java:6)

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.next();
        System.out.println("Enter your grades:");
        String grade = scan.next();
    }
}
```



OUTPUT

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

Examples:-  
Keyboard interrupts,



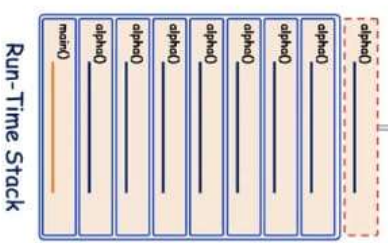
## Demonstration of Runtime Errors

### RECURSION

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

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

AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
alpha()  
AR for  
main()



Run-Time Stack

4

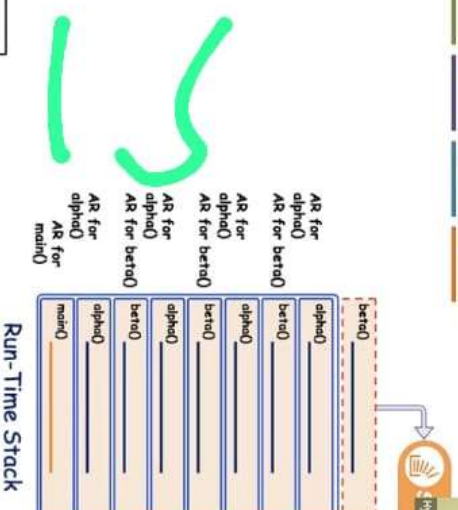
## Demonstration of Runtime Errors

### RECURSION

```
class Launch
{
    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)
```



## Demonstration of Runtime Errors



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

Impossible to

1000000000 \* 4 = 4 GB

continuously on the heap

OUTPUT

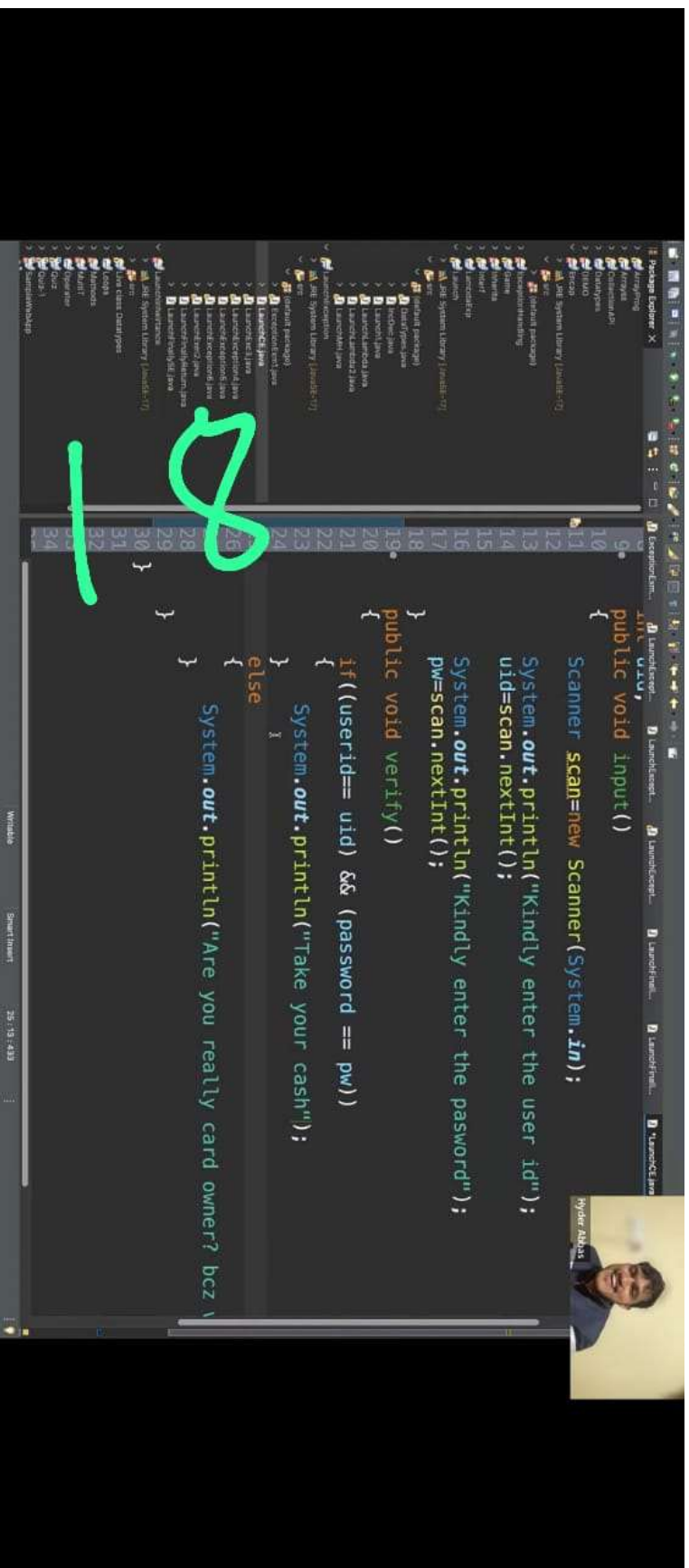
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space  
at launch.main(launch.java:5)

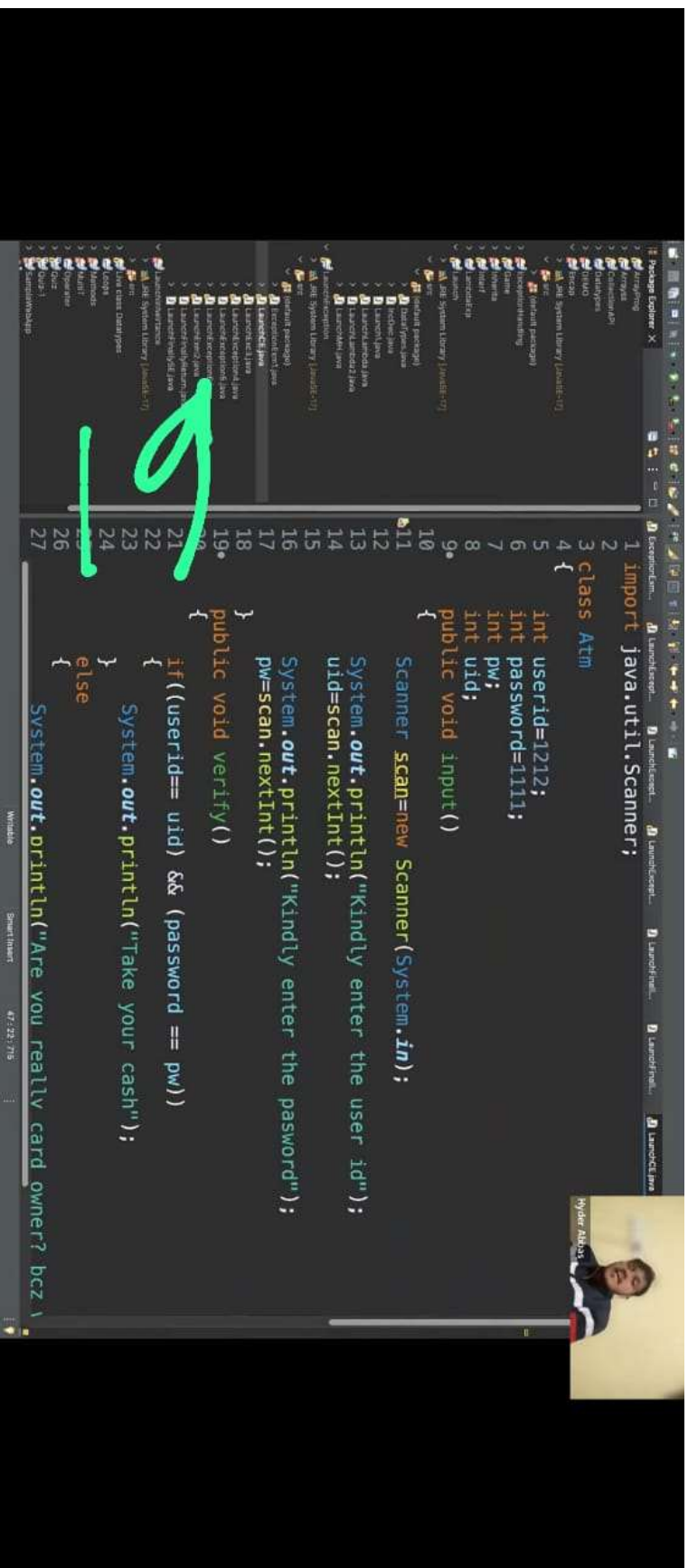
18



⇒ Custom Exception

17

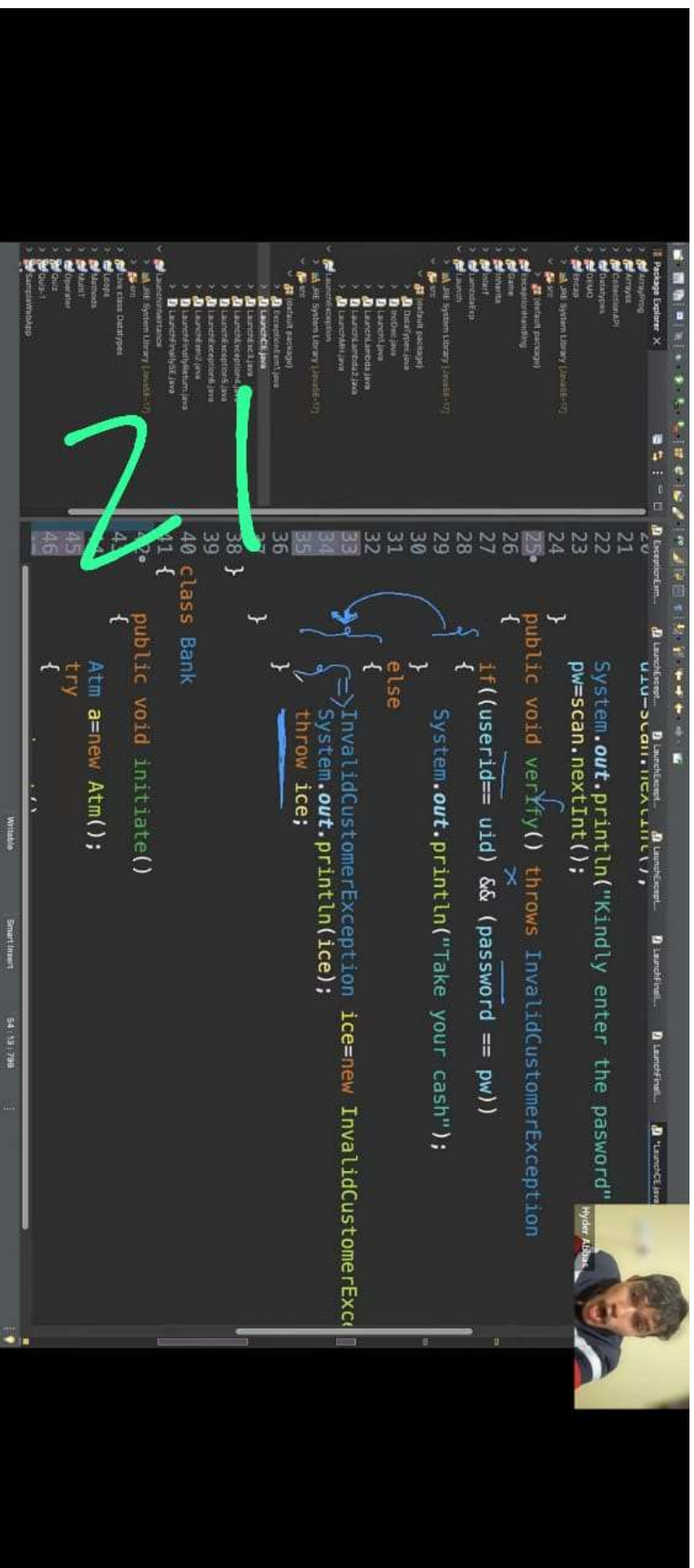


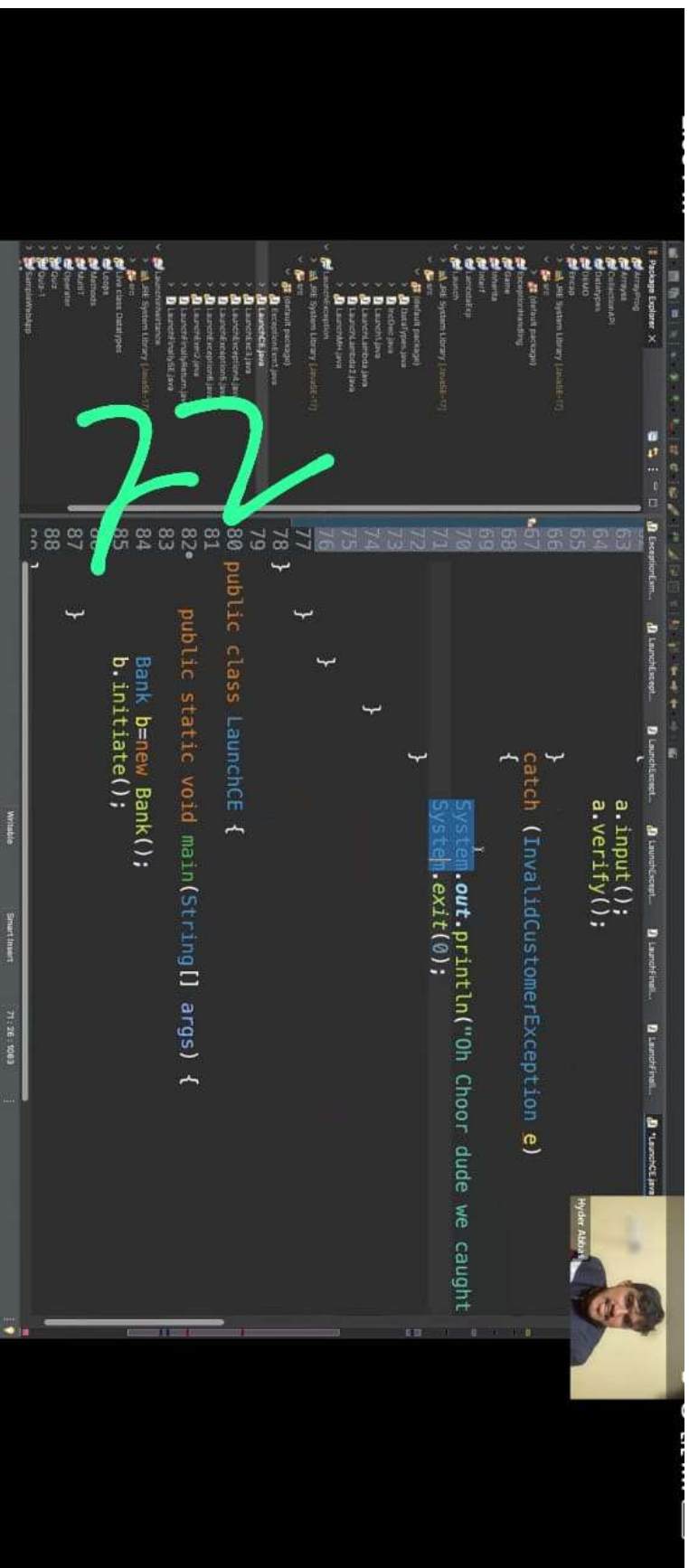


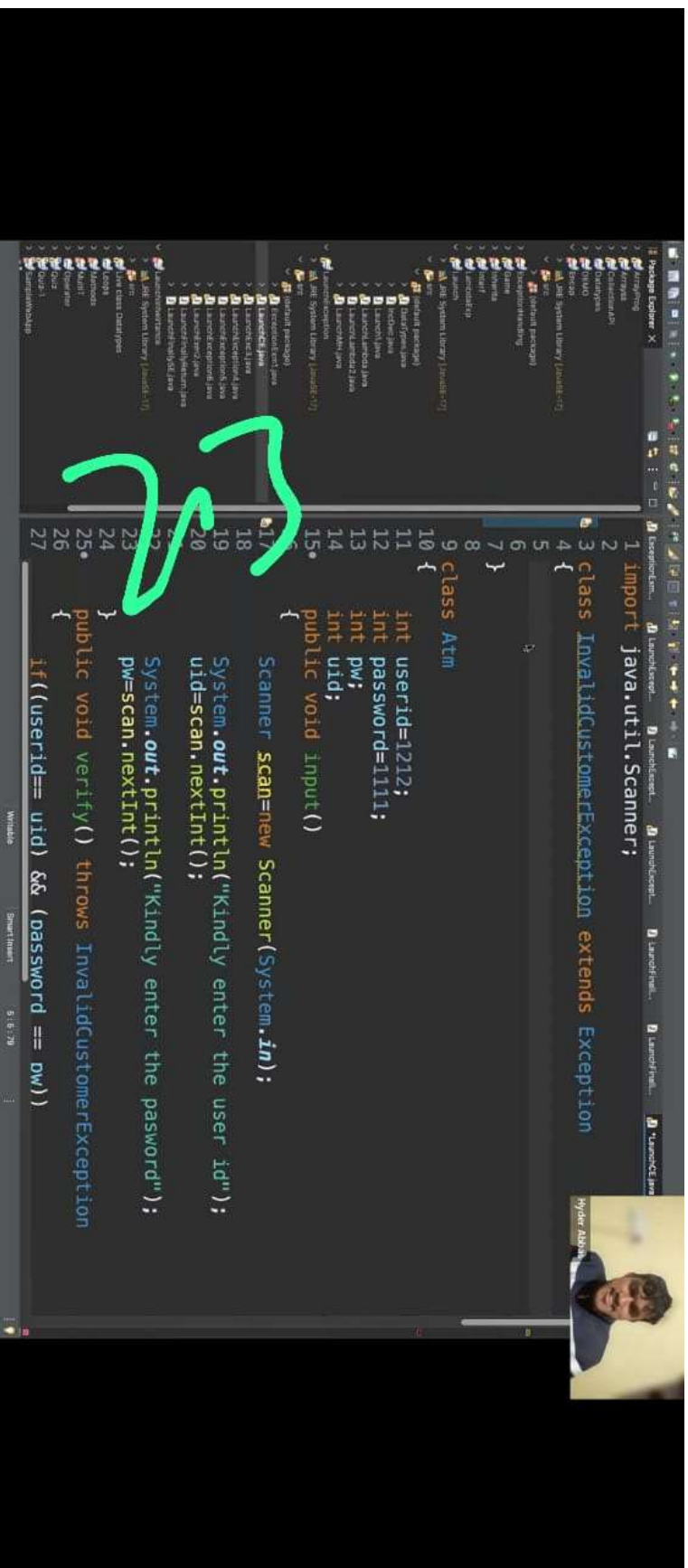
20

```
24 }
25 }
26 else
27 {
28     System.out.println("Are you really card owner?");
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         Bank b=new Bank();
46         b.initiate();
47     }
48 }
49
50
```





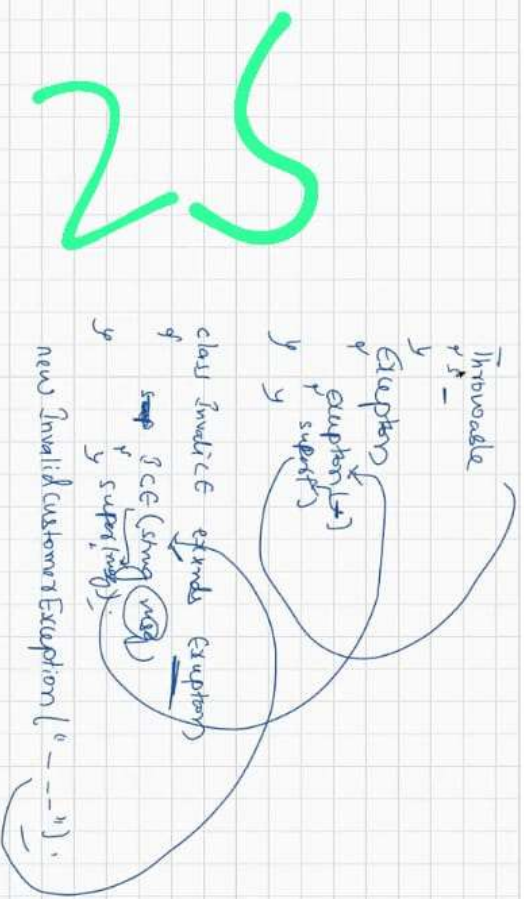












laptop X  
Hyd X  
Goo X  
Goo X  
13th X  
Goo X  
28th X  
Mys X  
Goo X  
Hyd X  
Dbr X  
Goo X  
Full X

keep.google.com/NOTES/186864470762.31931250

class Inrowable extends Object  
{  
    String msg;  
    public Inrowable (String msg)  
    {  
        this.msg = msg;  
    }  
    String getMsg() {  
        return msg;  
    }  
}

Hyder Abba

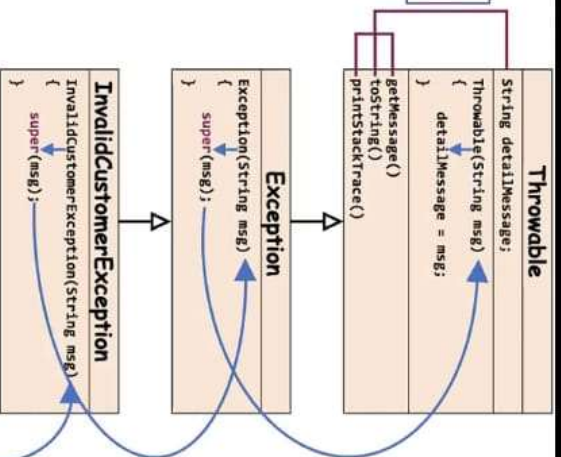
26

class Exception extends Throwable  
↳ support(msg).  
↳ class IOException extends Exception  
↳ IOException(msg).  
↳ IOException()

27



detailMessage is used by these methods to print exception information

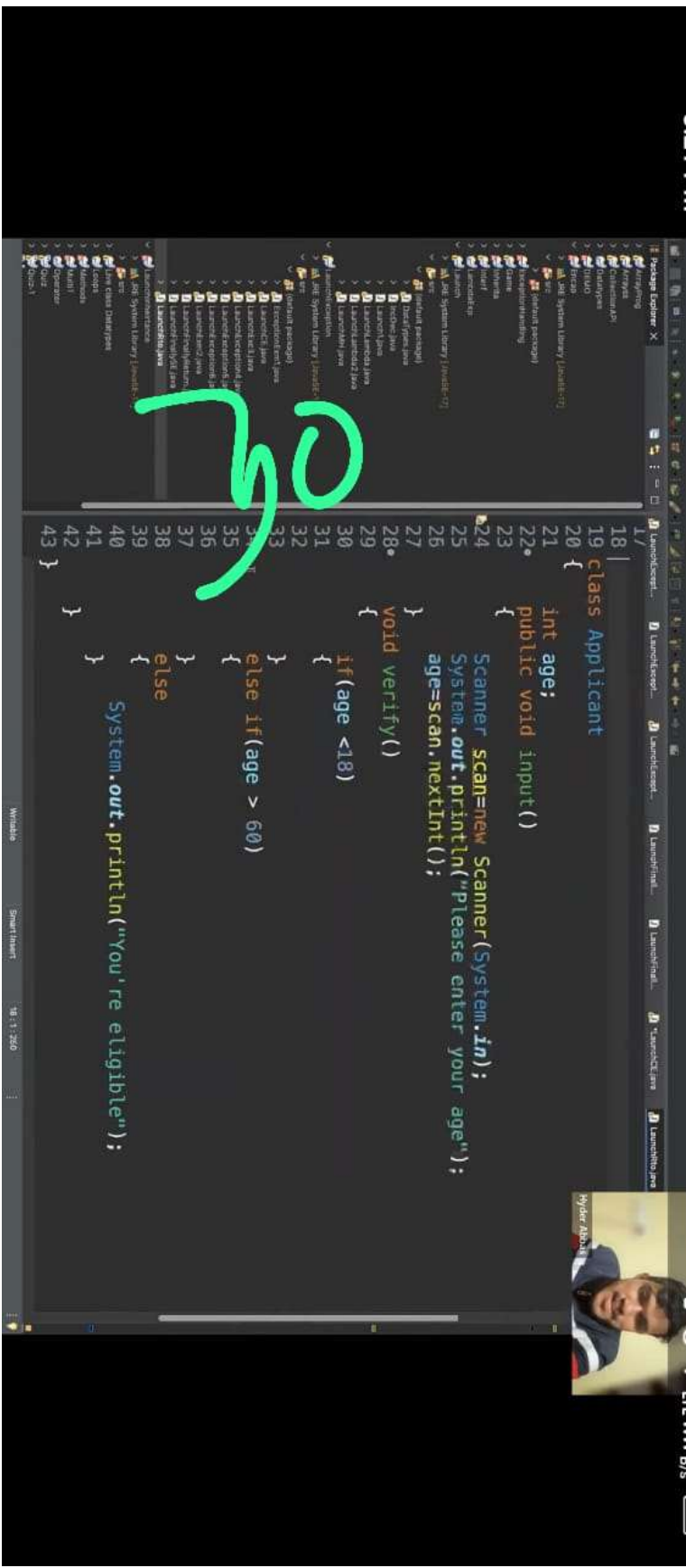


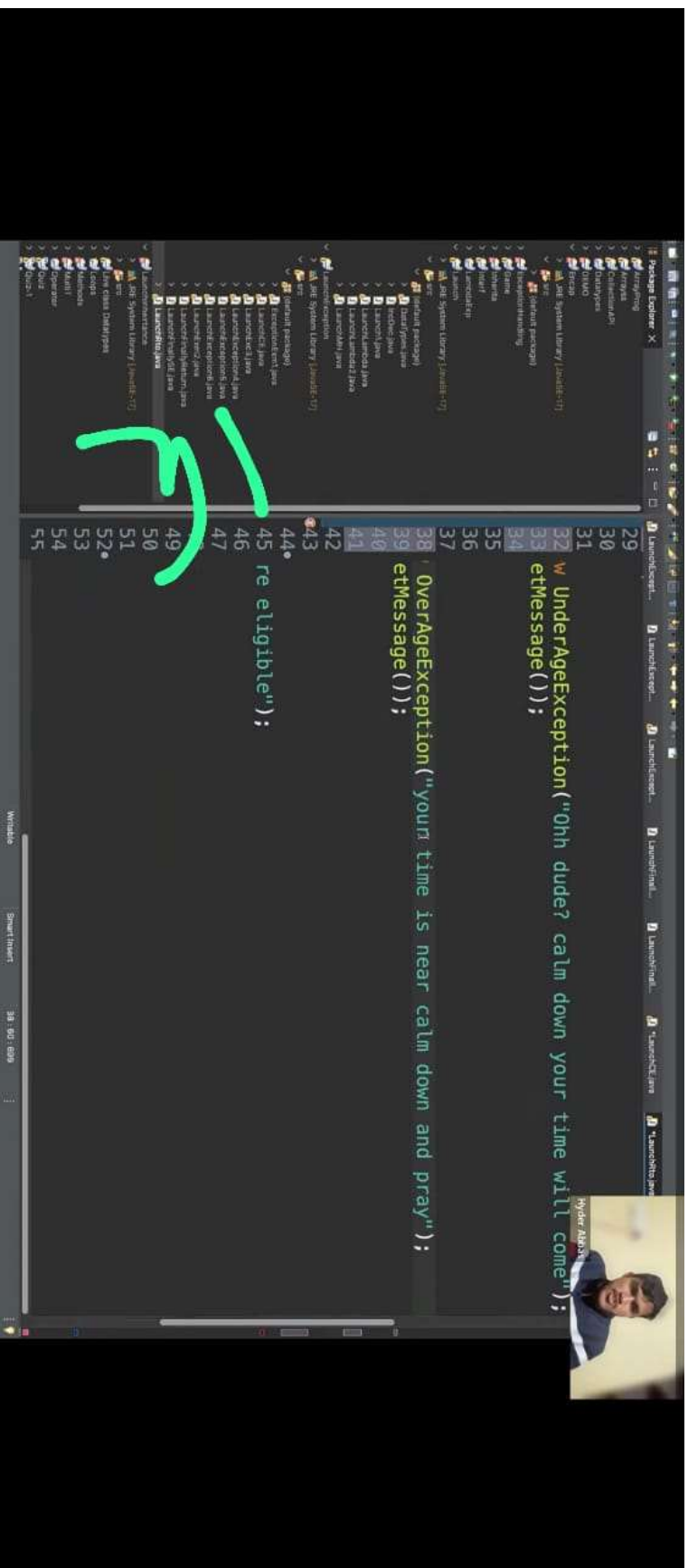
`new InvalidCustomerException("Invalid input. Try again!");`

28

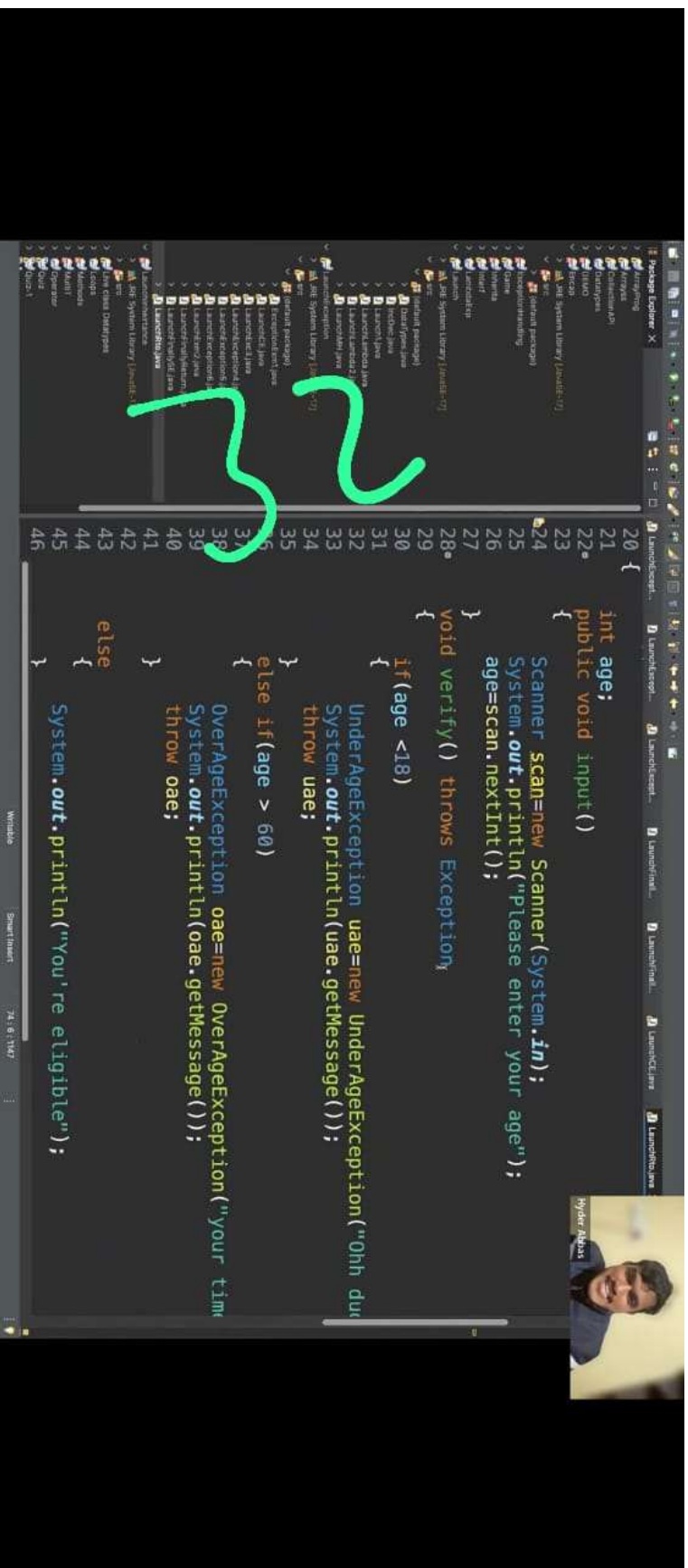




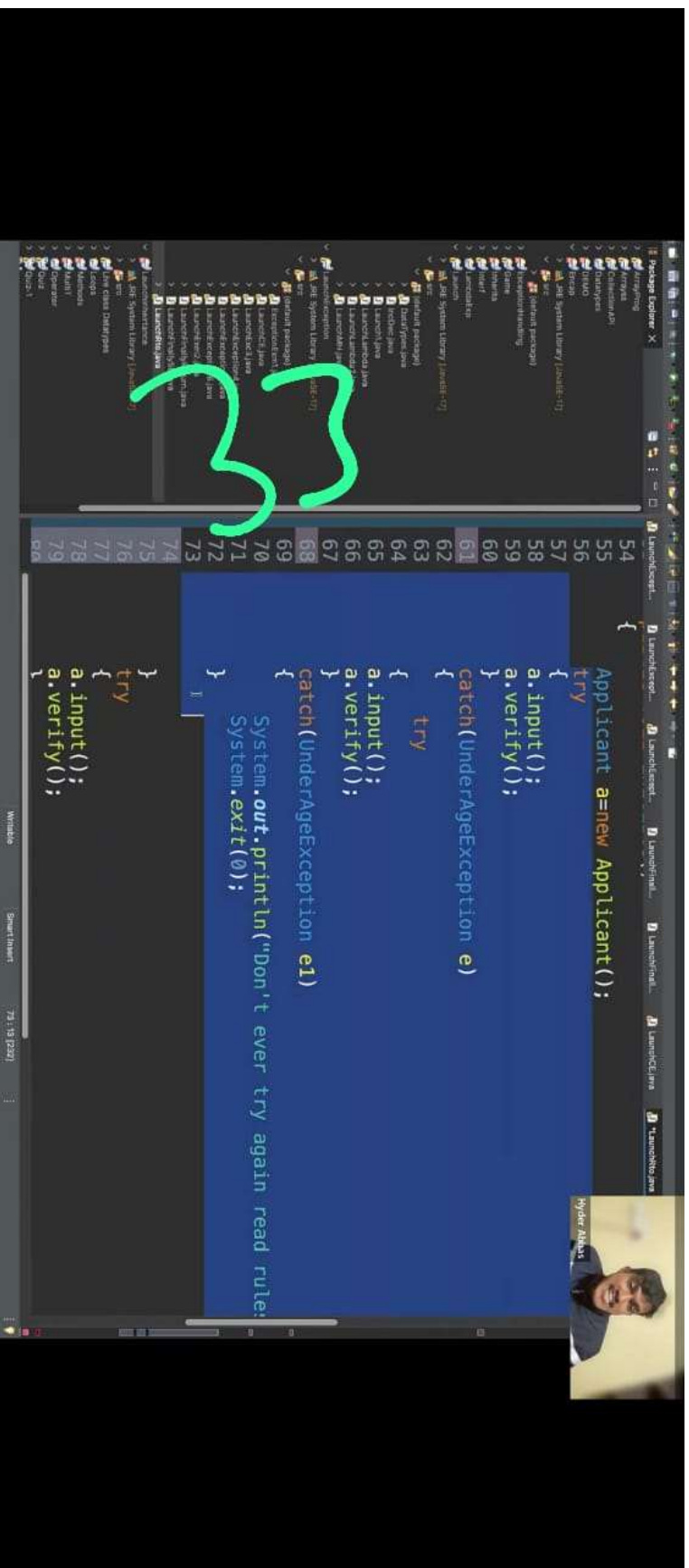


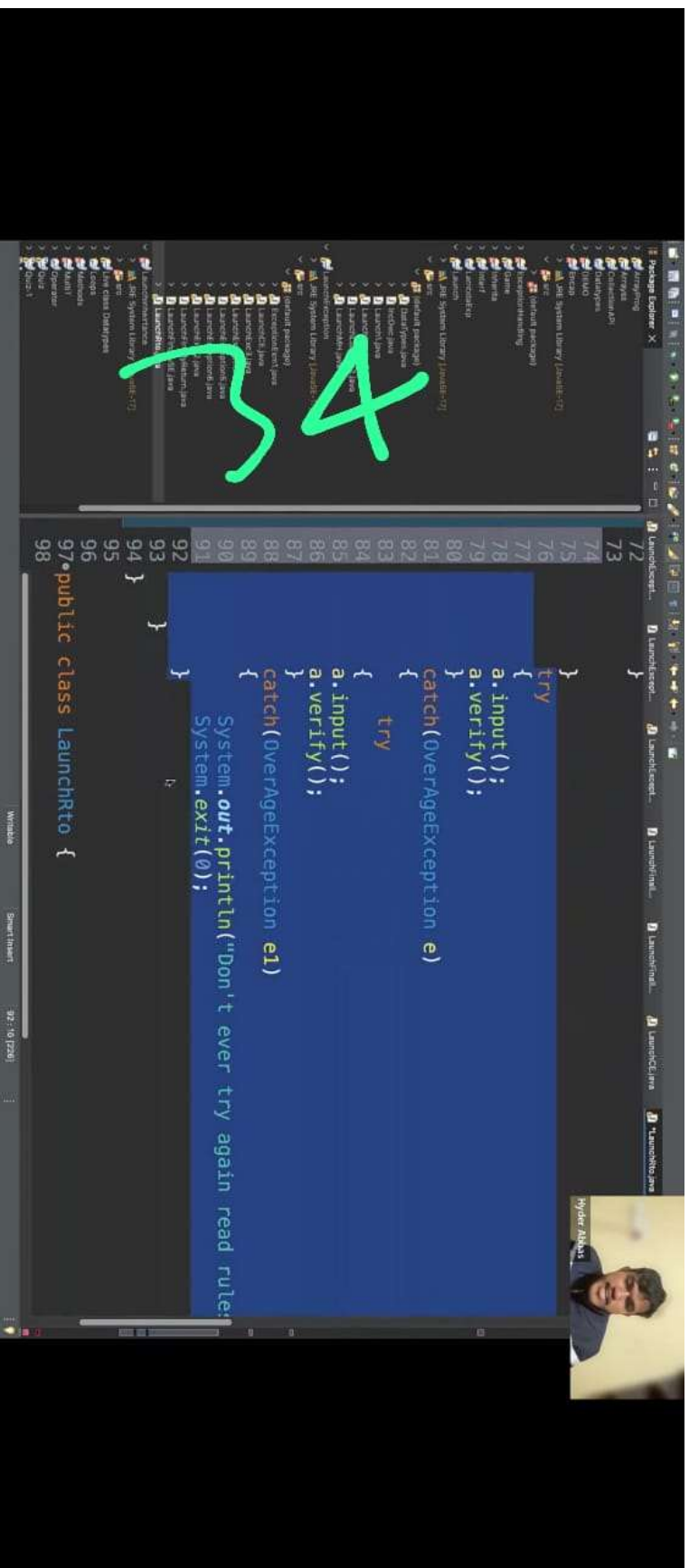














```
d:\code\... 2) Launching... 3) Launching... 4) Launching...  
d.input();  
a.verify();  
  
catch(underAgeException | overAgeException e)  
    try  
    {  
        a.input();  
        a.verify();  
    }  
    catch(overAgeException e)  
    {  
        try  
        {  
            System.out.println("Don't ever try again read rule:  
                System.exit(0);  
        }  
    }  
}  
  
}  
try  
{  
    a.input();  
    a.verify();  
}  
catch(overAgeException e)  
{  
    try
```



**Life • Ill B/s**



Syder Abbas

```
17 Please enter your age
17 Ohh dude? calm down your time will come
16 Please enter your age
16 Ohh dude? calm down your time will come
16 don't ever try again read rules
```

Untitled - Paint

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

Consolas

T4 B I U S

Background fill

Yasar Dargaznabi

```
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) {
        dt = new DeluxeThingy();
        dt.m.go();
        Thingy t = new DeluxeThingy();
        t.m.go();
    }
}
```

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

```
graph LR
    DT((DeluxeThingy)) -- dt --> DT2((DeluxeThingy))
    DT -- m --> M((Meter))
    M -- go() --> P[printing "m"]
```

672.57mm 657 x 109mm 3490 x 3885px

20°C Mostly clear

Q Search

Taskbar icons

100% 2313 29.11.2022

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

mc

```
//meter object would come and meter would extend component so navigation is possible
public static void main(String[] args) {
    DeluxeThing dt = new DeluxeThing();
    dt.m.go();
    Thingy t = new DeluxeThing();
    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. DeluxeThing is-a Component.
- F. DeluxeThing 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



Q>

```
public class Test{  
    public static void main(String[] args){
```

```
        boolean a =new Boolean(Boolean.valueOf(args[0]));  
        System.out.println(a);
```

```
        boolean b =new Boolean(args[1]);  
        System.out.println(b);  
    }  
}
```

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









Nishi Mahalingam



Nishit Chaudhary



Nishu Menon



Nithin Menon



Nishu Dhanrajani



Vigneshwaran




Niranjan Singh




Sidiq Mujahid



Mohan Dasgupta

- 
- Václav Klaus



Mohit Dasgupta

41

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);  
    }  
}
```

1

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

30



29.11.2022\_Jayprakash - Notepad

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



Ln 39, Col 1  
27°C  
Monday, 29 Nov

Q Search

100% Windows (CTRL) UTR 8  
FMS 33.36  
29.11.2022

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

1

49



28.11.2022, 10:07:00, images - Paint

File View

Clipboard Image Tools Brushes Shapes Size Colors

```
class Vehicle{
    int x;
    Vehicle(){
        this(10);
    }
    Vehicle(int x){
        this.x = x;
    }
}

int y;
Car(){
    super(); //line-n1
    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 y= new Car();
        System.out.println(y);
    }
}
```

45

1024x768px 3490 x 3865px 75.5KB

Q Search

100% 23.12 28.11.2022

28.11.2022, 10:07:00, images - Paint

File View

Clipboard Image Tools Brushes Shapes Size Colors

int y;

Car(){

super(); //line-n1

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 y= new Car();

System.out.println(y);

}

}

45

1024x768px 3490 x 3865px 75.5KB

Q Search

100% 23.12 28.11.2022

28.11.2022, 10:07:00, images - Paint

File View

Clipboard Image Tools Brushes Shapes Size Colors

int y;

Car(){

super(); //line-n1

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 y= new Car();

System.out.println(y);

}

}

45

1024x768px 3490 x 3865px 75.5KB

Q Search

100% 23.12 28.11.2022

```
    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 y = new Car();
        System.out.println(y);
    }
}
Predict the answer
A. 10:20
B. 0:20
C. Compilation fails at line n1
D. Compilation fails at line n2
Answer: D
```

40



\*28\_11\_2022\_omgpran\_cloveros - Notepad

File Edit View

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

Answer: D

Q>

```
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?

- A. three
- B. other
- 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.

Answer: A

4



Answer: A

Q&gt;

Given:

```
public class Foo {  
    static int[] a;  
    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

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

48





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 Dog, 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

43



```
...
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 Dog, Beagle is a Animal, Beagle has-a tail, Beagle is-A Jumper
41.     public void jump() {}
42. }
```

```
...
50. class Cat implements Jumper { // Cat is-a 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
- F. Beagle has-a Tail
- G. Beagle has-a Jumper

Answer: BCF

50



Given:

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. 13
- B. 134
- C. 1234
- D. 2134
- E. 2143
- F. 4321

Answer: C(1234)

IS

