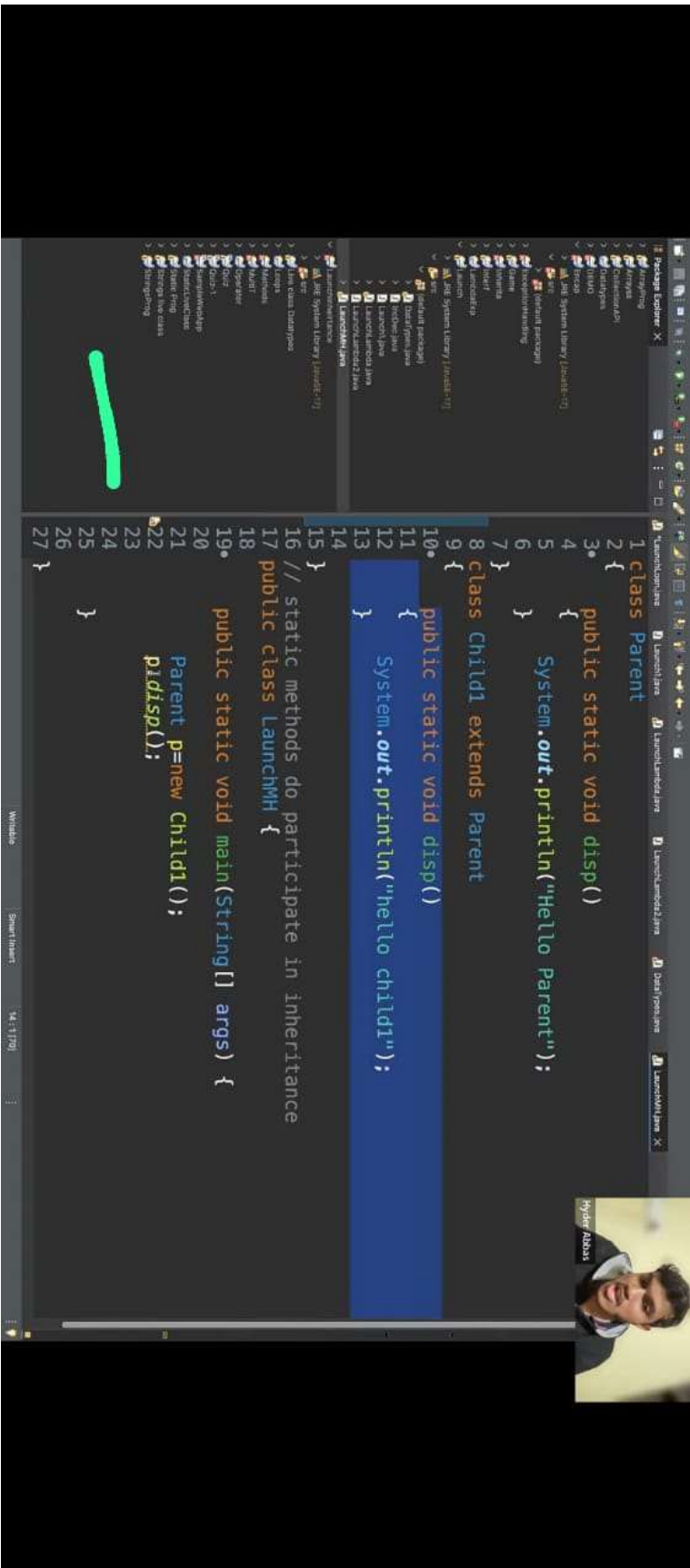


# JAVA EXCEPTION



The screenshot shows an IDE window with the following Java code:

```
1 class Parent
2 {
3     public static void disp()
4     {
5         System.out.println("Hello Parent");
6     }
7 }
8 class Child1 extends Parent
9 {
10     public static void disp()
11     {
12         System.out.println("hello child1");
13     }
14 }
15 }
16 // static methods do participate in inheritance
17 public class LaunchMH {
18
19     public static void main(String[] args) {
20
21         Parent p=new Child1();
22         p.disp();
23     }
24 }
25
26
27 }
```

A video call overlay is visible in the bottom right corner, showing a person with the name "Syder Abbas" below their video feed.

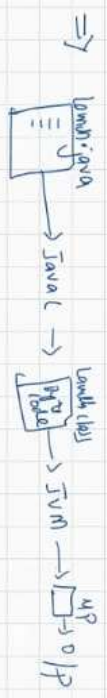
Browser tabs: chrome.pdf, Hyder Abbas, keep.google.com/NOTIFY6836437456.426129549, Learn Data, Google Meet, 20th Nov, MySQL, Google Meet

Hyder Abbas

1. Exception Handling

⇒ Exception Handling

Java Pgm → Running

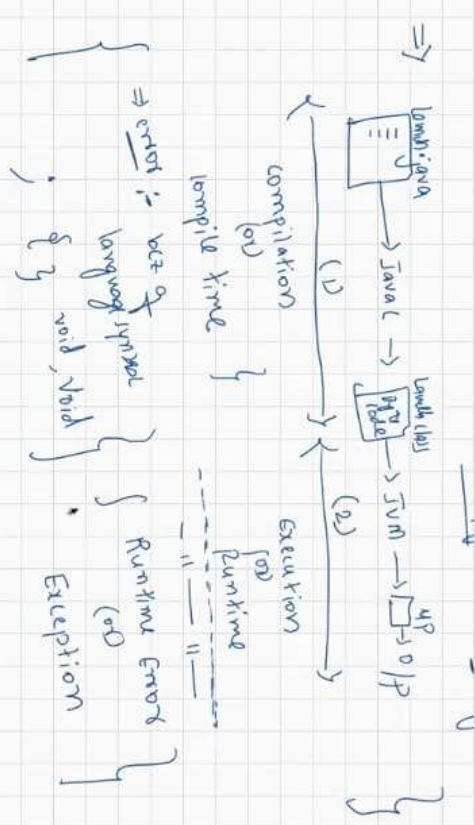


S

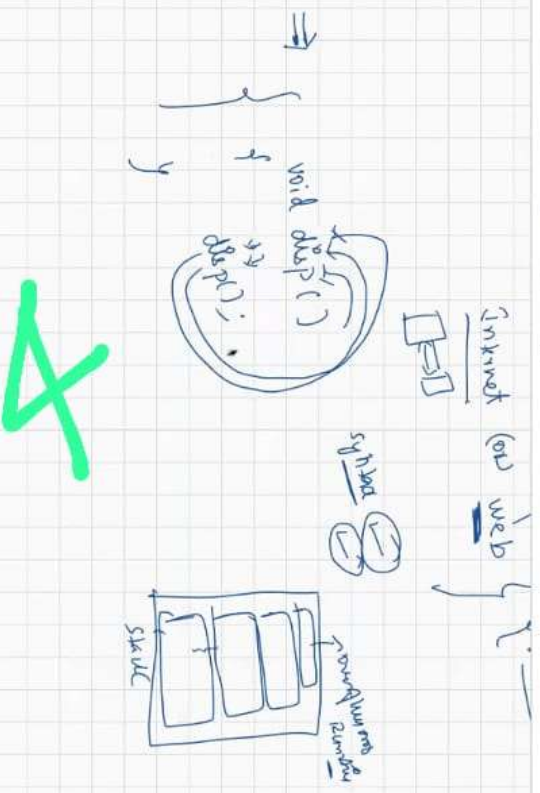
Task list: X Hyper-Admin: X Google Meet: X Google Meet: X Learn Data: X Google Meet: X 20th Nov: X MySQL: X Google Meet: X

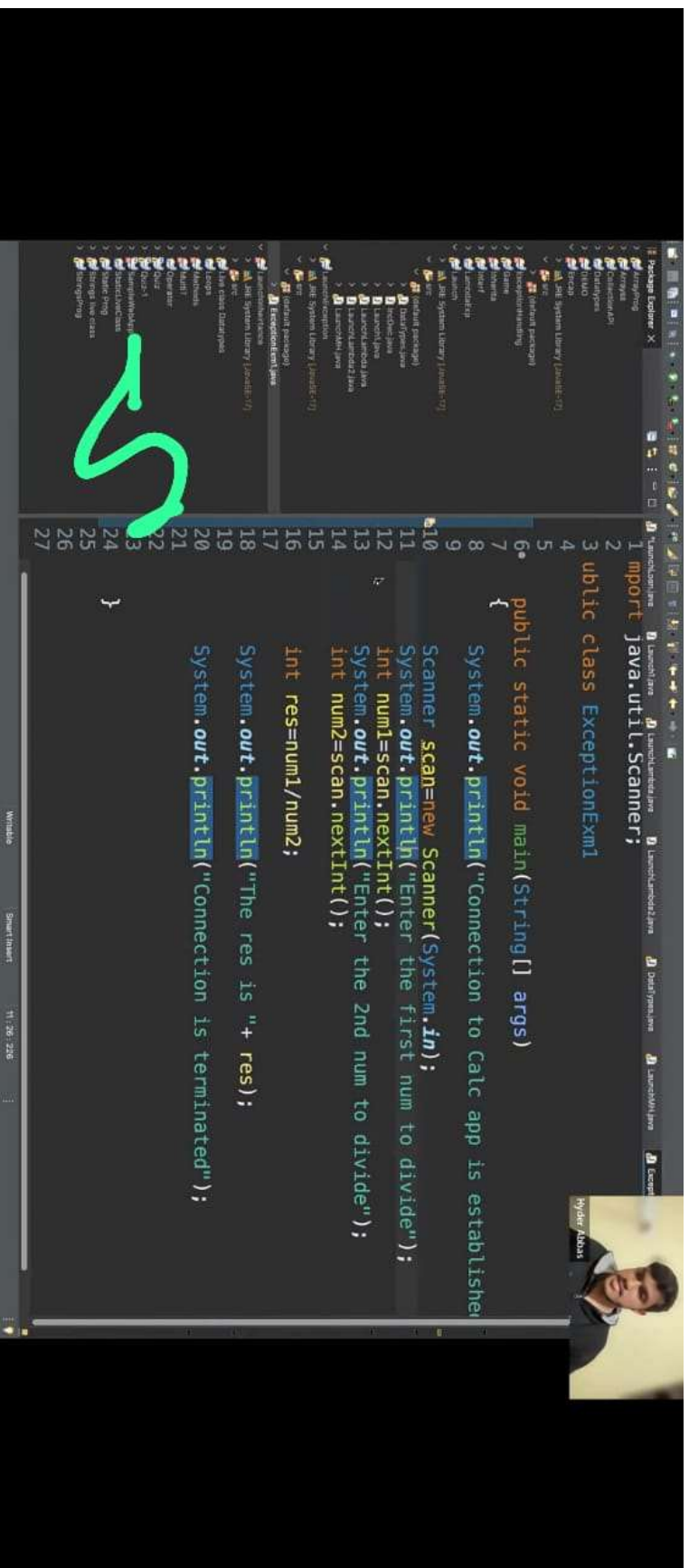
keep.google.com/NOTIFY8838437456426129549

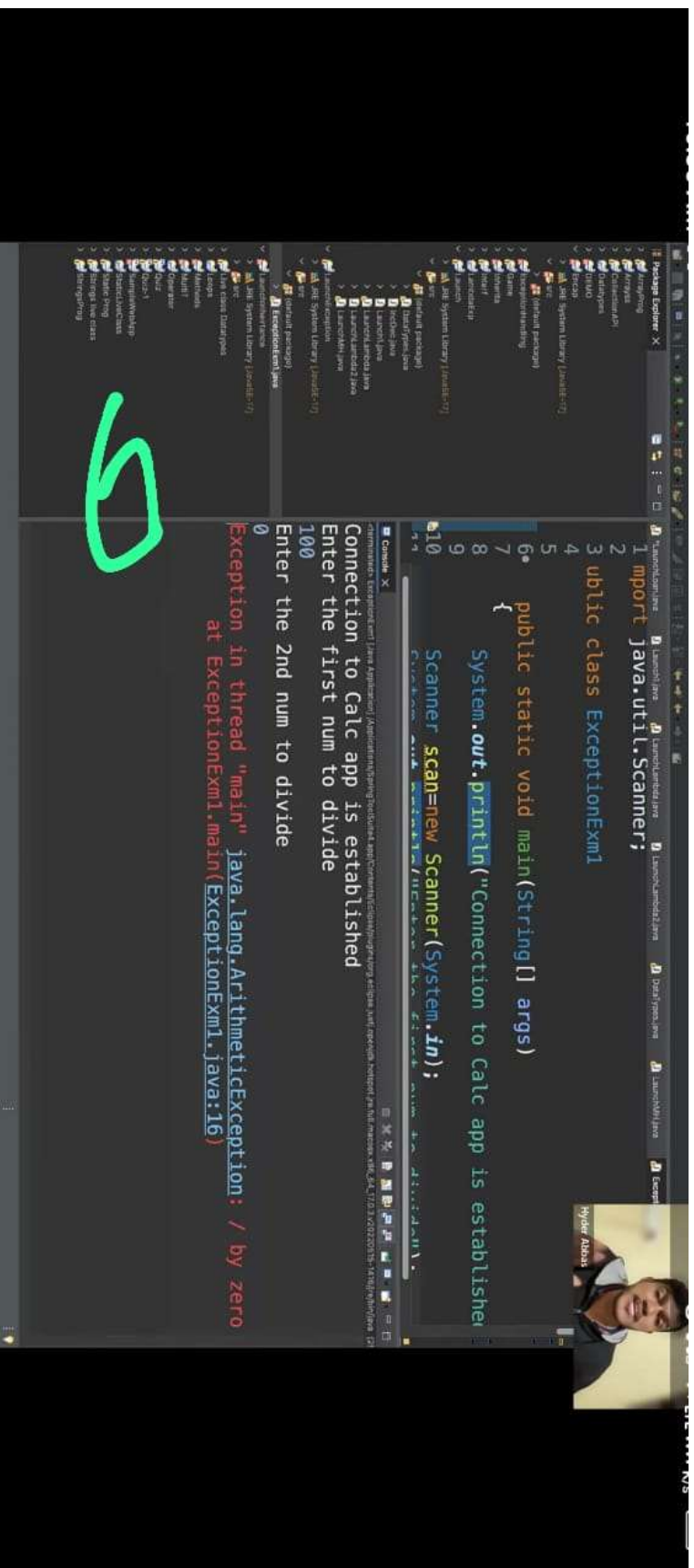
Hyder Abbas



3



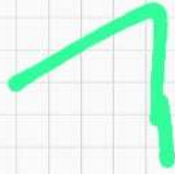




$\Rightarrow$  Exception Hierarchy  $\xrightarrow{1}$  Inbuilt class  $\Rightarrow$  using we can handle exception  
 $\Rightarrow$  checked exception  $\{$  unchecked exception

$\Rightarrow$  chemodectoma

unchecked exception





Hyder Abbas

```
60 public static void main(String[] args)
61 {
62     System.out.println("Connection to Calc app is
63     try
64     {
65         Scanner scan=new Scanner(System.in);
66         System.out.println("Enter the first num to divide");
67         int num1=scan.nextInt();
68         System.out.println("Enter the 2nd num to divide");
69         int num2=scan.nextInt();
70         int res=num1/num2;
71         System.out.println("The res is "+ res);
72     }
73     catch(Exception e)
74     {
75         System.out.println("You're trying to divide by zero
76         System.out.println("Connection is terminated");
77     }
78 }
79 }
80 }
81 }
```

22:10:42



11:08 AM

The screenshot shows an IDE with a Java file named `LaunchAndDiv.java`. The code is as follows:

```
6 public static void main(String[] args)
7 {
8     System.out.println("Connection to Calc app is
9     try
10     {
11         Scanner scan=new Scanner(System.in);
12         System.out.println("Enter the first num to divide");
13         int num1=scan.nextInt();
14         System.out.println("Enter the 2nd num to divide");
15         int num2=scan.nextInt();
```

The console output shows the following sequence of events:

```
Connection to Calc app is established
Enter the first num to divide
100
Enter the 2nd num to divide
0
You're trying to divide by zero illogical
connection is terminated
```

A red handwritten mark is visible in the bottom right corner of the IDE window.



⇒ If programme has handled the except will matter to

Output ∈ H? (No)

⇒ catch ... )

q/w/y

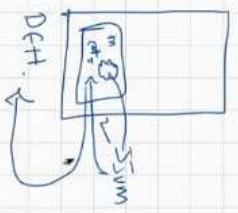
Exception

- ↳ Primitive Exception
- ↳ Negative Array Exception
- ↳ ArrayIndexOutOfBoundsException

10



⇒



→ HT1 max (0.9999)  $\approx$   
 → Negative bias (0.0001)  
 → Negative bias (0.0001)

=

11:23 AM

```
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

System.out.println("The res is " + res);

System.out.println("Enter the size of an array");
int size=scan.nextInt();
int[] a=new int[size];

System.out.println("Enter the elem to be inserted in that array");
int elem=scan.nextInt();

System.out.println("Enter the pos at which elem has to be inserted");
int pos=scan.nextInt();

a[pos]=elem;
System.out.println("Element " + elem + " Inserted at " + pos +" successfully");
}
catch(Exception e)
{
    System.out.println("Wrong input please give right one");
}
System.out.println("Connection is terminated");
}
```



✓✓

```

25 System.out.println("Enter the elem to be inserted in that array");
26 int elem=scan.nextInt();
27
28 System.out.println("Enter the pos at which elem has to be inserted");
29 int pos=scan.nextInt();
30
31 a[pos]=elem;
32 System.out.println("Element " + elem + " Inserted at "+ pos +" successfully");
33
34 }
35 catch(ArithmeticException ae)
36 {
37     System.out.println("Please provide non zero denominator");
38 }
39 catch(NegativeArraySizeException nae)
40 {
41     System.out.println("Please be positive!");
42 }
43 catch(ArrayIndexOutOfBoundsException a)
44 {
45     System.out.println("Be in your limits, Don't cross it");
46 }
47
48 System.out.println("Connection is terminated");
49
50 }
51 }

```

✓

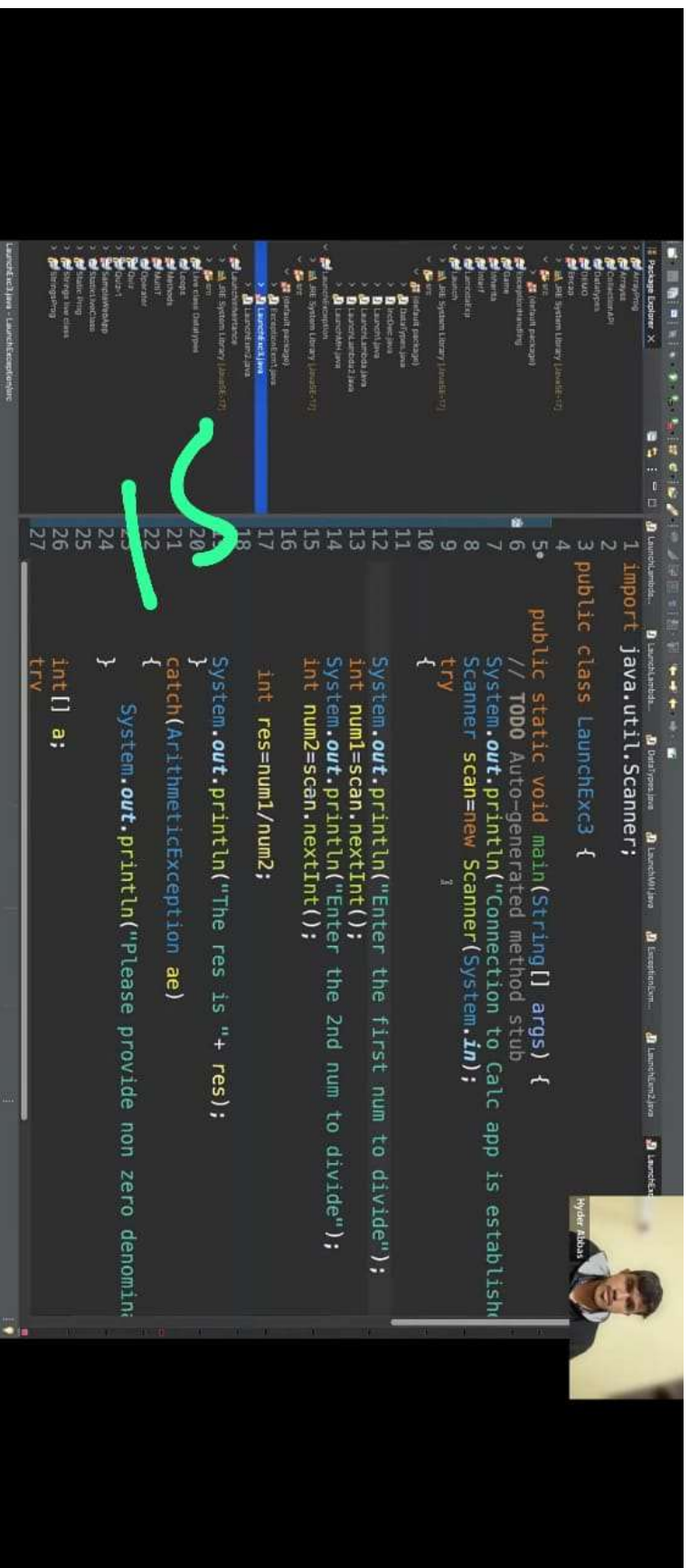


```
29 int pos=scan.nextInt();
30
31 a[pos]=elem;
32 System.out.println("Element " + elem + " Inserted at " + pos + " successfully");
33
34
35 }
36 catch(ArithmeticException ae)
37 {
38     System.out.println("please provide non zero denominator");
39 }
40 catch(NegativeArraySizeException nae)
41 {
42     System.out.println("please be positive!");
43 }
44 catch(ArrayIndexOutOfBoundsException a)
45 {
46     System.out.println("Be in your limits, Don't cross it");
47 }
48 catch(Exception e)
49 {
50     System.out.println("Wrong input");
51 }
52
53 System.out.println("Connection is terminated");
54 }
```

✓







```
1 import java.util.Scanner;
2
3 public class LaunchExc3 {
4
5     // TODO Auto-generated method stub
6     public static void main(String[] args) {
7         Scanner scan=new Scanner(System.in);
8         try
9         {
10
11             System.out.println("Enter the first num to divide");
12             int num1=scan.nextInt();
13             System.out.println("Enter the 2nd num to divide");
14             int num2=scan.nextInt();
15
16             int res=num1/num2;
17
18             System.out.println("The res is " + res);
19         }
20         catch(ArithmeticException ae)
21         {
22             System.out.println("Please provide non zero denominator");
23         }
24     }
25
26     int[] a;
27     try
```



```

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

    int size=scanner.nextInt();
    a=new int[size];

    catch(NegativeArraySizeException nae)
    {
        System.out.println("please be positive!");
    }

    try
    {
        System.out.println("Enter the elem to be inserted in that array");
        int elem=scanner.nextInt();

        System.out.println("Enter the pos at which elem has to be inserted");
        int pos=scanner.nextInt();

        a[pos]=elem;
        System.out.println("Element "+ elem + " Inserted at "+ pos + " successfully");
    }

    catch(ArrayIndexOutOfBoundsException aa)
    {
        System.out.println("Be in your limits, Don't cross it");
    }

    catch(Exception e)
    {

```

16



```

38 System.out.println("Enter the pos at which elem has to be inserted")
39 int pos=scanner.nextInt();
40
41
42 a[pos]=elem;
43 System.out.println("Element "+ elem + " Inserted at "+ pos +" successfully");
44
45 }
46 catch(NegativeArraySizeException nae)
47 {
48     System.out.println("be positive");
49 }
50
51 catch(ArrayIndexOutOfBoundsException aa)
52 {
53     System.out.println("be in your limits, Don't cross it");
54 }
55 catch(Exception e)
56 {
57     System.out.println("Wrong input!");
58 }
59
60 System.out.println("Connection is terminated");
61
62 }
63

```

✓



```
1 import java.util.Scanner;
2
3 public class LaunchExc3 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner scan=new Scanner(System.in);
8         try
9         {
10
11             System.out.println("Enter the first num to divide");
12             int num1=scan.nextInt();
13             System.out.println("Enter the 2nd num to divide");
14             int num2=scan.nextInt();
15
16             int res=num1/num2;
17
18             System.out.println("The res is "+ res);
19         }
20         catch(ArithmeticException ae)
21         {
22             System.out.println("Please provide non zero denominator");
23         }
24
25
26
27     }
28 }
```

18



File Edit View

25.11.2022, 21:00:00:0000 - Notepad

Given:

```
1. public class Barn {  
2.     public static void main(String[] args) {  
3.         new Barn().go("hi", 1);  
4.         new Barn().go("hi", "world", 2);  
5.     }  
6.     public void go(String... y, int x) {  
7.         System.out.print(y[y.length - 1] + "  
8.     }  
9. }
```

What is the result?

- A. hi hi
- B. hi world
- C. world world
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: D

19



Ln 20, Col 1

27°C  
Partly cloudy



100%

Windows [CTRL]

UTZ 5

ENG  
IN  
25.11.2022  
22:09

Answer: D

Q>

What is the result?

1. public class Person {
12.     String name = "No name";
13.     public Person(String nm) { name = nm; }
14. }
- 15.
16. public class Employee extends Person {
17.     String empID = "0000";
18.     public Employee(String id) { empID = id; }
19. }
- 20.
21. public class EmployeeTest {
22.     public static void main(String[] args){
23.         Employee e = new Employee("4321");
24.         System.out.println(e.empID);
25.     }
26. }

20



1. public class Mud {
2. //insert code here
3. System.out.println("hi");
4. }
5. }

And the following five fragments:

```
public static void main(String...a) {  
    public static void main(String * a) {  
    public static void main(String... a) {  
    public static void main(String[]... a) {  
    public static void main(String...[] a) {
```

How many of the code fragments, inserted independently at line 2, compile?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5

Answer: D

32





```
4. class Rock extends Atom {  
5.   Rock(String type) { super(); System.out.print(type); }  
6. }  
7. public class Mountain extends Rock {  
8.   Mountain() {  
9.     super("granite");  
10.    new Rock("granite");  
11.  }  
12.   public static void main(String[] a) { new Mountain(); }  
13. }
```

What is the result?

- A. Compilation fails.
- B. atom granite
- C. granite granite
- D. atom granite granite
- E. An exception is thrown at runtime.
- F. atom granite atom granite

Answer: F (constructor chaining) atom granite atom granite

23



### Question

```
interface TestA { String toString(); }  
public class Test {  
    public static void main(String[] args) {  
        System.out.println(new TestA() { // creating an object for a class which implements TestA interface  
            public String toString() { return "test"; }  
        });  
    }  
}
```

What is the result?

- A. test
- B. null
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 1.
- E. Compilation fails because of an error in line 4.
- F. Compilation fails because of an error in line 5.

test



Given:

- 11. abstract class Vehicle { public int speed() { return 0; } }
- 12. class Car extends Vehicle { public int speed() { return 60; } }
- 13. class RaceCar extends Car { public int speed() { return 150; } ... }

21. RaceCar racer = new RaceCar();

22. Car car = new RaceCar();

23. Vehicle vehicle = new RaceCar();

24. System.out.println(racer.speed() + ", " + car.speed() + ", " + vehicle.speed());

What is the result?

- A. 0, 0, 0
- B. 150, 60, 0
- C. Compilation fails.
- D. 150, 150, 150
- E. An exception is thrown at runtime.

Vehicle

|=> int speed() => 0

Car

25



E. An exception is thrown at runtime.

Vehicle

|=> int speed() => 0

|

Car

|=> int speed() => 60

|

RaceCar (object)

|=> int speed() => 150

Answer: D(becoz of overriding jvm will use the runtime object)150,150,150 |

26



## Question

```
5. class Building {}  
6. public class Barn extends Building {  
7.     public static void main(String[] args) {  
8.         Building build1 = new Building();  
9.         Barn barn1 = new Barn();  
10.        Barn barn2 = (Barn) build1;  
11.        Object obj1 = (Object) build1;  
12.        String str1 = (String) build1;  
13.        Building build2 = (Building) barn1;  
14.    }  
15. }
```

Which is true?

- A. If line 10 is removed, the compilation succeeds.
- B. If line 11 is removed, the compilation succeeds.
- C. If line 12 is removed, the compilation succeeds.
- D. If line 13 is removed, the compilation succeeds.
- E. More than one line must be removed for compilation to succeed.



```
11.      Object obj1 = (Object) build1;  
12.      String str1 = (String) build1; //RE: ClassCastException  
13.      Building build2 = (Building) barn1;  
14.      }  
15. }
```

Which is true?

- A. If line 10 is removed, the compilation succeeds.
- B. If line 11 is removed, the compilation succeeds.
- C. If line 12 is removed, the compilation succeeds.
- D. If line 13 is removed, the compilation succeeds.
- E. More than one line must be removed for compilation to succeed.

Object ==> obj1

Building ==> build1

Barn ==> barn1

Answer: C

28









Nabeel Manjrahi

70

- 22°C  
Partly cloudy

 Search

22

25

22.52  
25-11-2002