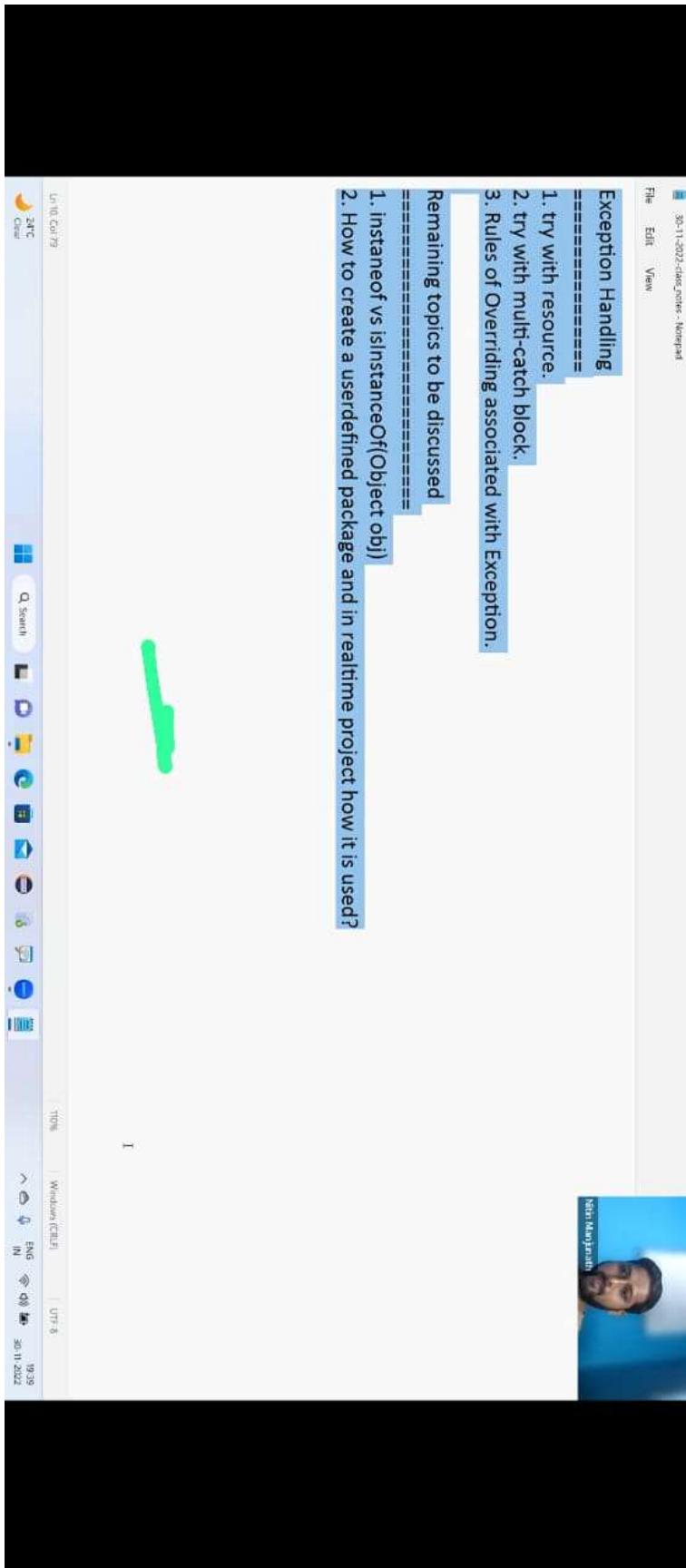


Java Exception Part 4



30-11-2022-class-prime - Notepad

File Edit View

Exception Handling

1. try with resource.
2. try with multicatch block.
3. Rules of Overriding associated with Exception.

Remaining topics to be discussed

1. instanceof vs isInstanceOf(Object obj)
2. How to create a userdefined package and in realtime project how it is used?

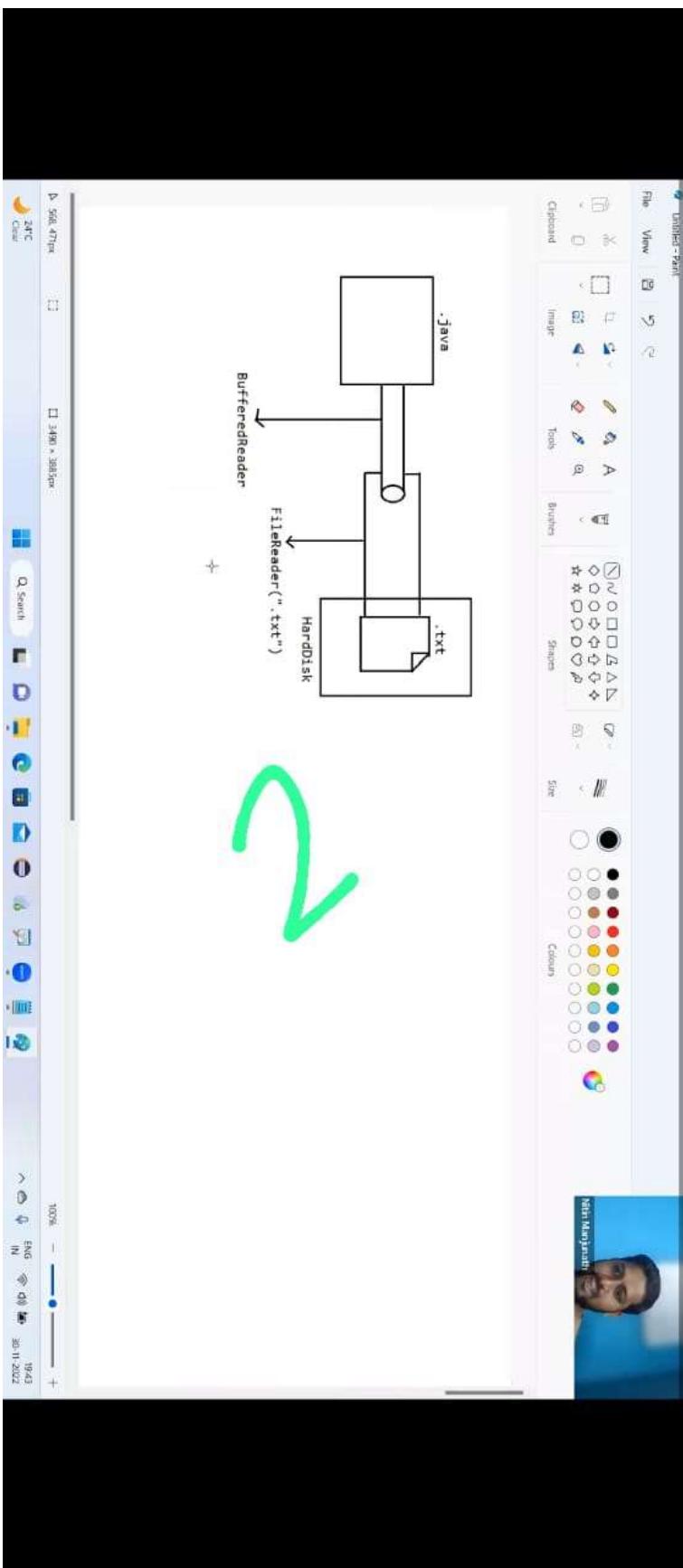
I

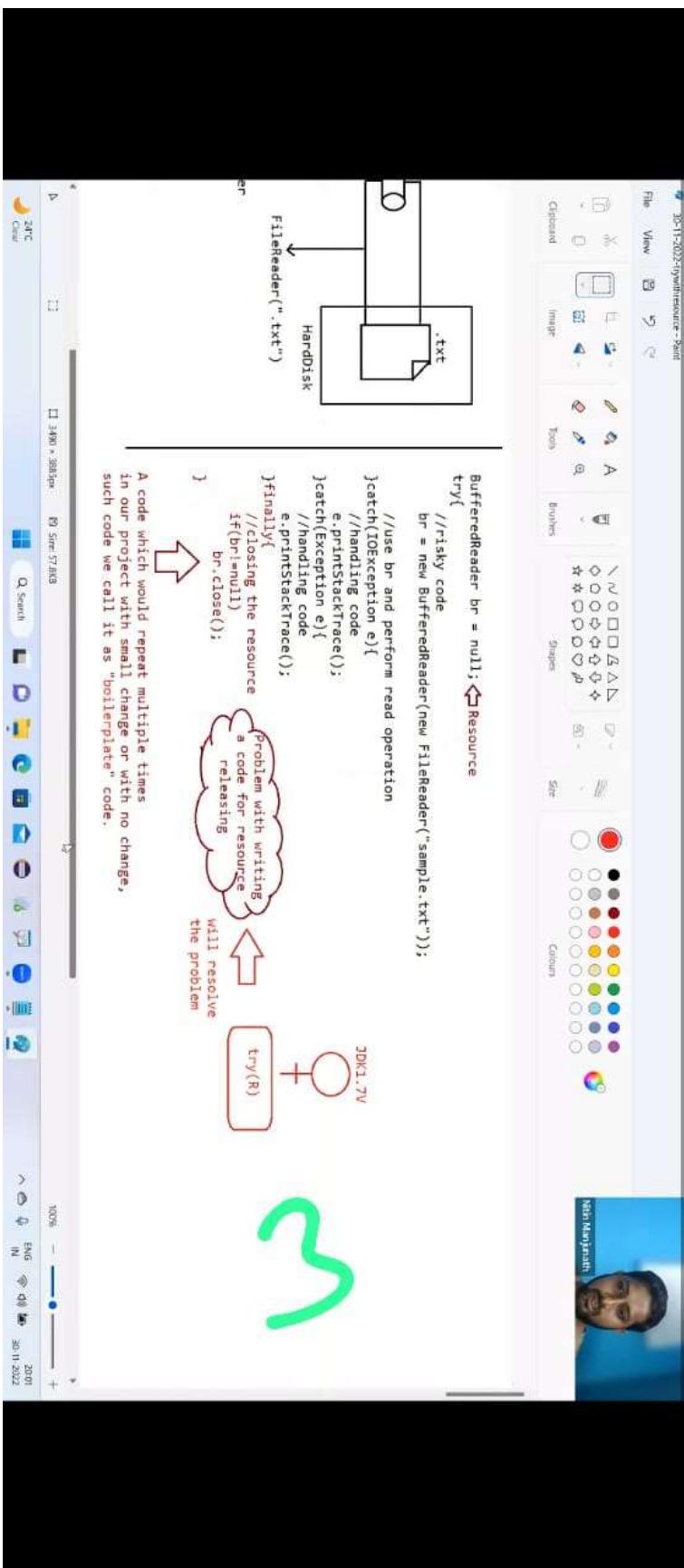
100% Windows (CEP) UTF-8

24°C Chat

Search

10:10 AM 30 Nov 2022





The screenshot shows a presentation slide with a background image of a person speaking. The slide contains the following text and code:

trywithresource ✓

```
try(BufferedReader br = new BufferedReader(new FileReader("sample.txt"))){  
    //use br based on ur requirement  
    // automatically the br will be close once the control reaches to end of try either normally  
    // or abnormally we are not required to close explicitly  
}
```

JDK1.7V

At this line br will be closed automatically

```
catch(IOException e){  
    //handling code  
    e.printStackTrace();  
}  
catch(Exception e){  
    //handling code  
    e.printStackTrace();  
}
```

times
- with no change,
- "e" code.

4

Annotations and highlights:

- A red checkmark is placed next to the word "trywithresource".
- A red circle with a plus sign is placed next to the text "JDK1.7V".
- A red bracket groups the first two lines of code under the heading "Not a Good approach".
- A red bracket groups the last three lines of code under the heading "will resolve the problem".



```
30-11-2022-class-prime - Notepad  
File Edit View  
until jdk1.6, it is compulsorily required to write finally block to close all the resources which are open as a [Edit Margins]  
  
eg:: BufferedReader br=null  
try{  
    br=new BufferedReader(new FileReader("abc.txt"));  
}catch(IOException ie){  
    ie.printStackTrace();  
}finally{  
    try{  
        if(br!=null){  
            br.close();  
        }  
    }catch(IOException ie){  
        ie.printStackTrace();  
    }  
}
```

Problems in the approach

File Edit View

Problems in the approach

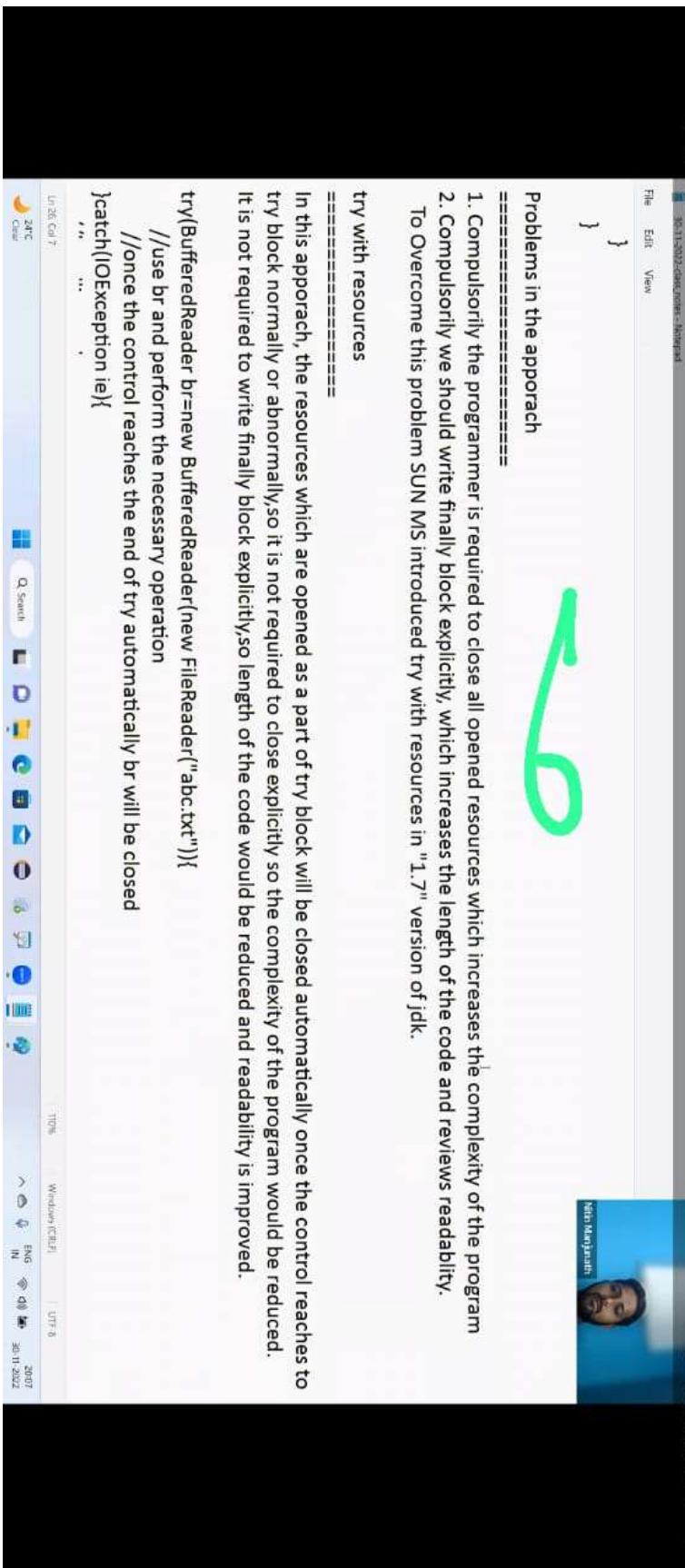
1. Compulsorily the programmer is required to close all opened resources which increases the complexity of the program
 2. Compulsorily we should write finally block explicitly, which increases the length of the code and reviews readability.

To Overcome this problem SUN MS introduced try with resources in "1.7" version of jdk.

try with resources

In this approach, the resources which are opened as a part of try block will be closed automatically once the control reaches to try block normally or abnormally, so it is not required to close explicitly so the complexity of the program would be reduced. It is not required to write finally block explicitly, so length of the code would be reduced and readability is improved.

```
try(BufferedReader br=new BufferedReader(new FileReader("abc.txt")){  
    //use br and perform the necessary operation  
    //once the control reaches the end of try automatically br will be closed  
}catch(IOException ie){  
    ...  
}
```



- 30.11.2022 class-prime - Notepad
- File Edit View
1. Compulsorily the programmer is required to close all opened resources which increases the complexity of the code.
 2. Compulsorily we should write finally block explicitly, which increases the length of the code and reviews.
- To Overcome this problem SUN MS introduced try with resources in "1.7" version of jdk.

try with resources

In this approach, the resources which are opened as a part of try block will be closed automatically once the control reaches to try block normally or abnormally, so it is not required to close explicitly so the complexity of the program would be reduced. It is not required to write finally block explicitly, so length of the code would be reduced and readability is improved.

```
try(BufferedReader br=new BufferedReader(new FileReader("abc.txt"))){  
    //use br and perform the necessary operation  
    //once the control reaches the end of try automatically br will be closed  
}catch(IOException ie){  
    //handling code  
}
```

7



```
D:\Wrapper\classes>javac Test.java
Test.java:5: error: incompatible types: try-with-resources not applicable to variable
      try(String s =new String("sachin")){
                         ^
      (String cannot be converted to AutoCloseable)
1 error

D:\Wrapper\classes>javap java.lang.AutoCloseable
Compiled from "AutoCloseable.java"
public interface java.lang.AutoCloseable {
    public abstract void close() throws java.lang.Exception;
}

D:\Wrapper\classes>
```

9

```
9.0.1 AVI Command Prompt < + >

1 error

D:\Wrapper\classes>javap java.lang.AutoCloseable
Compiled from "AutoCloseable.java"
public interface java.lang.AutoCloseable {
    public abstract void close() throws java.lang.Exception;
}

D:\Wrapper\classes>javap java.lang.String
Compiled from "String.java"
Error: Main method not found in class java.lang.String. Please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application

D:\Wrapper\classes>javap java.lang.String
public final class java.lang.String implements java.io.Serializable, java.lang.Comparable<java.lang.String>, java.lang.CharSequence {
    public static final java.util.Comparator<java.lang.String> CASE_INSENSITIVE_ORDER;
    public java.lang.String();
    public java.lang.String(java.lang.String);
    public java.lang.String(char[]);
    public java.lang.String(char[], int, int);
    public java.lang.String(byte[], int, int);
    public java.lang.String(byte[], int, int, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.lang.String(byte[], int, int, java.nio.charset.Charset);
    public java.lang.String(byte[], java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.lang.String(byte[], java.nio.charset.Charset);
    public java.lang.String(byte[], int, int);
    public java.lang.String(StringBuffer);
    public java.lang.String(java.lang.StringBuffer);
    public java.lang.String(char[], boolean);
    public int length();
    public boolean isEmpty();
    public char charAt(int);
    public int codePointAt(int);
    public int codePointsBefore(int);
    public int codePointCount(int, int);
    public int offsetByCodePoints(int, int);
}
```

0:02

Command Prompt

```
D:\Wrapper classes>javap java.io.BufferedReader
Compiled from "BufferedReader.java"
public class java.io.BufferedReader extends java.io.Reader {
    public java.io.BufferedReader(java.io.Reader, int);
    public java.io.BufferedReader(char[], int, int) throws java.io.IOException;
    public int read() throws java.io.IOException;
    public int read(char[], int, int) throws java.io.IOException;
    public java.lang.String readLine(boolean) throws java.io.IOException;
    public java.lang.String readLine() throws java.io.IOException;
    public long skip(long) throws java.io.IOException;
    public boolean ready() throws java.io.IOException;
    public boolean markSupported();
    public void mark(int) throws java.io.IOException;
    public void reset() throws java.io.IOException;
    public void close() throws java.io.IOException;
    public java.util.stream.Stream<java.lang.String> lines();
}
53: D:\Wrapper classes>
```

1 0

-2:43:18

2.0X

20/11/2022

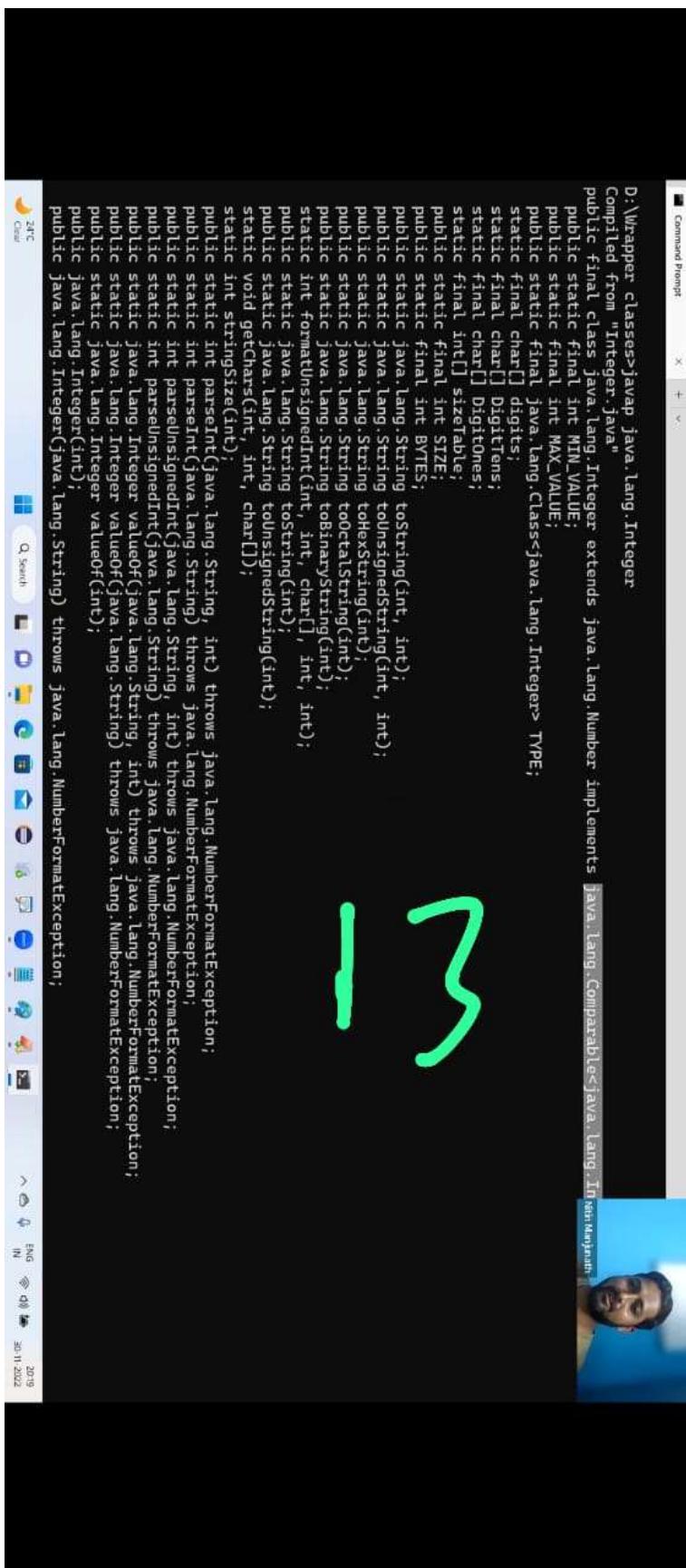
```
D:\Wrapper\classes>javap java.sql.Connection
Compiled from "Connection.java"
public interface java.sql.Connection extends java.sql.Wrapper,java.lang.AutoCloseable {
    public static final int TRANSACTION_NONE;
    public static final int TRANSACTION_READ_UNCOMMITTED;
    public static final int TRANSACTION_READ_COMMITTED;
    public static final int TRANSACTION_REPEATABLE_READ;
    public static final int TRANSACTION_SERIALIZABLE;
    public abstract java.sql.Statement createStatement() throws java.sql.SQLException;
    public abstract java.sql.PreparedStatement prepareStatement(java.lang.String) throws java.sql.SQLException;
    public abstract java.sql.CallableStatement prepareCall(java.lang.String) throws java.sql.SQLException; public abstract void setAutoCommit(boolean) throws java.sql.SQLException;
    public abstract boolean getAutoCommit() throws java.sql.SQLException;
    public abstract void commit() throws java.sql.SQLException;
    public abstract void rollback() throws java.sql.SQLException;
    public abstract void close() throws java.sql.SQLException;
    public abstract boolean isClosed() throws java.sql.SQLException;
    public abstract java.sql.DatabaseMetaData getMetaData() throws java.sql.SQLException;
    public abstract void setReadOnly(boolean) throws java.sql.SQLException;
    public abstract boolean isReadOnly() throws java.sql.SQLException;
    public abstract void setCatalog(java.lang.String) throws java.sql.SQLException;
    public abstract java.lang.String getCatalog() throws java.sql.SQLException;
    public abstract void setTransactionIsolation(int) throws java.sql.SQLException;
    public abstract int getTransactionIsolation() throws java.sql.SQLException;
    public abstract java.sql.SQLWarning getWarnings() throws java.sql.SQLException;
    public abstract void clearWarnings() throws java.sql.SQLException;
    public abstract java.sql.Statement createStatement(int, int) throws java.sql.SQLException;
```



D:\Wrapper\classes>javap java.sql.Statement

Compiled from "Statement.java"

```
public interface java.sql.Statement extends java.sql.Wrapper, java.lang.AutoCloseable {
    public static final int CLOSE_CURRENT_RESULT;
    public static final int KEEP_CURRENT_RESULT;
    public static final int CLOSE_ALL_RESULTS;
    public static final int SUCCESS_NO_INFO;
    public static final int EXECUTE_FAILED;
    public static final int RETURN_GENERATED_KEYS;
    public static final int NO_GENERATED_KEYS;
    public abstract java.sql.ResultSet executeQuery(java.lang.String) throws java.sql.SQLException;
    public abstract int executeUpdate(java.lang.String) throws java.sql.SQLException;
    public abstract void close() throws java.sql.SQLException;
    public abstract int getMaxFieldSize() throws java.sql.SQLException;
    public abstract void setMaxFieldSize(int) throws java.sql.SQLException;
    public abstract int getMaxRows() throws java.sql.SQLException;
    public abstract void setMaxRows(int) throws java.sql.SQLException;
    public abstract void setEscapeProcessing(boolean) throws java.sql.SQLException;
    public abstract int getQueryTimeout() throws java.sql.SQLException;
    public abstract void setQueryTimeout(int) throws java.sql.SQLException;
    public abstract void cancel() throws java.sql.SQLException;
    public abstract java.sql.SQLWarning getWarnings() throws java.sql.SQLException;
    public abstract void clearWarnings() throws java.sql.SQLException;
    public abstract void setCursorName(java.lang.String) throws java.sql.SQLException;
    public abstract boolean execute(java.lang.String) throws java.sql.SQLException;
    public abstract java.sql.ResultSet getResultSet() throws java.sql.SQLException;
    public abstract int getUpdateCount() throws java.sql.SQLException;
    public abstract boolean getMoreResults() throws java.sql.SQLException;
}
```



D:\Wrapper\classes>javap java.util.Scanner

Compiled from "Scanner.java"

```
public final class java.util.Scanner implements java.util.Iterator<java.lang.String>, java.io.Closeable {
    static final boolean $assertionsDisabled;
    public java.util.Scanner(java.lang.Readable);
    public java.util.Scanner(java.io.InputStream);
    public java.util.Scanner(java.io.InputStreamReader, java.lang.String);
    public java.util.Scanner(java.io.File) throws java.io.FileNotFoundException;
    public java.util.Scanner(java.io.File, java.lang.String) throws java.io.FileNotFoundException;
    public java.util.Scanner(java.nio.file.Path) throws java.io.IOException;
    public java.util.Scanner(java.nio.file.Path, java.lang.String) throws java.io.IOException;
    public java.util.Scanner(java.lang.String);
    public java.util.Scanner(java.nio.channels.ReadableByteChannel);
    public java.util.Scanner(java.nio.channels.ReadableByteChannel, java.lang.String);
    public void close();
    public java.io.IOException IOException();
    public java.util.regex.Pattern delimiter();
    public java.util.Scanner useDelimiter(java.util.regex.Pattern);
    public java.util.Scanner useDelimiter(java.lang.String);
    public java.util.Locate locale();
    public java.util.Scanner useLocale(java.util.locale);
    public int radix();
    public java.util.Scanner useRadix(int);
    public java.util.regex.MatchResult match();
    public java.lang.String toString();
    public boolean hasNext();
    public java.lang.String next();
    public void remove();
    public boolean hasNext(java.lang.String);
    public java.lang.String next(java.lang.String);
    public boolean hasNext(java.util.regex.Pattern);
```

14

```
30.11.2022 class-prime - Notepad  
File Edit View  
}  
  
Nitin Manjrekar
```

Rules of using try with resource

1. we can declare any no of resources, but all these resources should be seperated with ;

eg#1.

```
try(R1;R2;R3){  
    //use the resources  
}
```

2. All resources are said to be AutoCloseable resources iff the class implements an interface called "java.lang.AutoCloseable"

either directly or indirectly

e.g.: java.io package classes, java.sql package classes

```
Compiled from "AutoCloseable.java"  
public interface java.lang.AutoCloseable {  
    public abstract void close() throws java.lang.Exception;  
}
```

IS

Line 70 Col 2 100% Windows (CEP) UTF-8
24C Clear

File Edit View

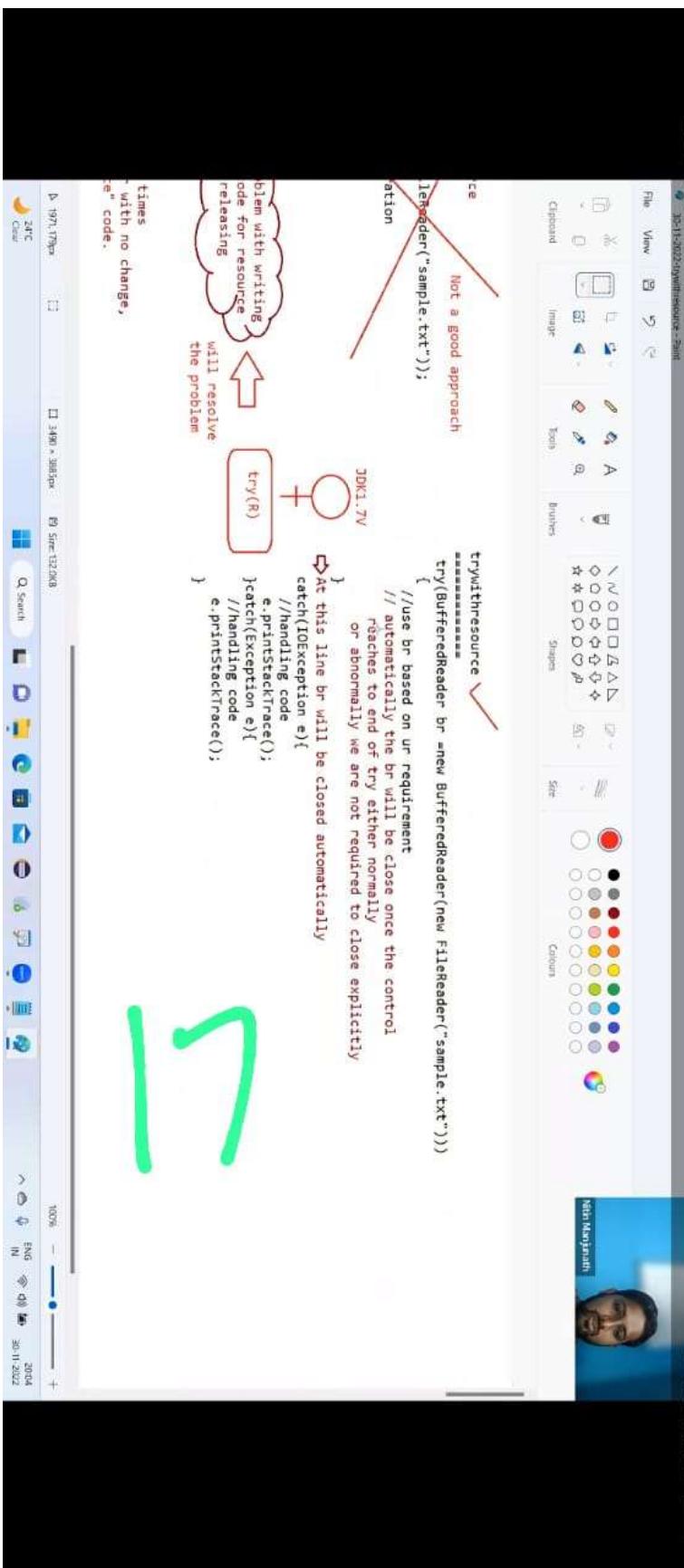
```
eg#1.
try(R1,R2;R3;){
    //use the resources
}
```

2. All resources are said to be AutoCloseable resources iff the class implements an interface called "java.lang.AutoCloseable" either directly or indirectly

eg:: java.io package classes, java.sql.package classes

```
public interface java.lang.AutoCloseable {
    public abstract void close() throws java.lang.Exception;
}
```

Note: whichever class has implemented this interface those classes objects are referred as "resources".



File Edit View Search Document Project Tools Browser Editor Window Help

Directory Object Functions

(D) New Volume D:\ WinRAR classes

Test.java

```
1 import java.io.*;
2
3 public class Test {
4     public static void main(String[] args) {
5
6         try(BufferedReader br = new BufferedReader(new FileReader("sample.txt"))){
7             br =new BufferedReader(new FileReader("output.txt"));
8         }
9         //br.close() will execute automatically and resource will be closed
10        catch(Exception e){
11            e.printStackTrace();
12        }
13    }
14 }
15
16
17
18
19
20
21
```

Java (*.java)

For Help press F1

24C C:\

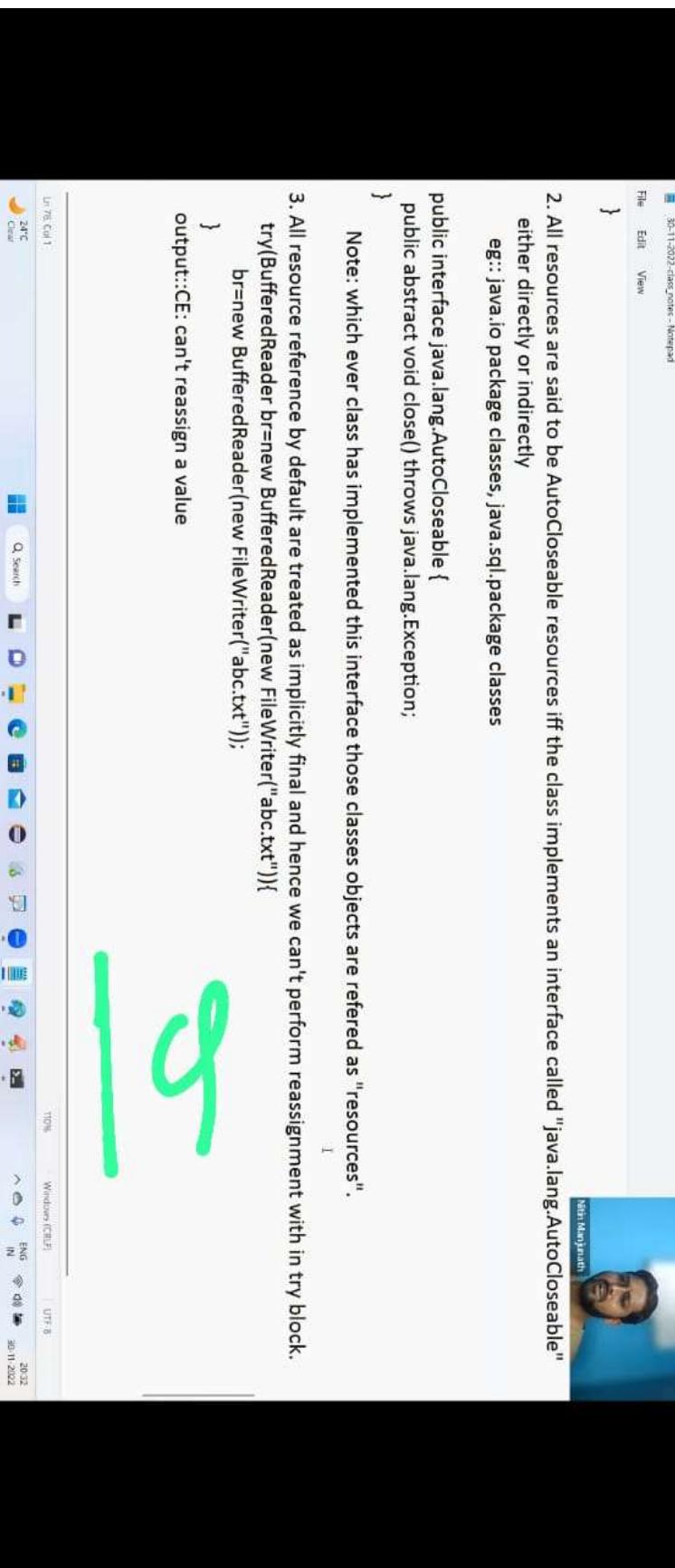
Q. Search

In 13 col 10 0 00 PC Absl

^ < > << >> <<< >>> <<<< >>>>

2011-11-2022

28



30.11.2022 class-prime - Notepad

File Edit View

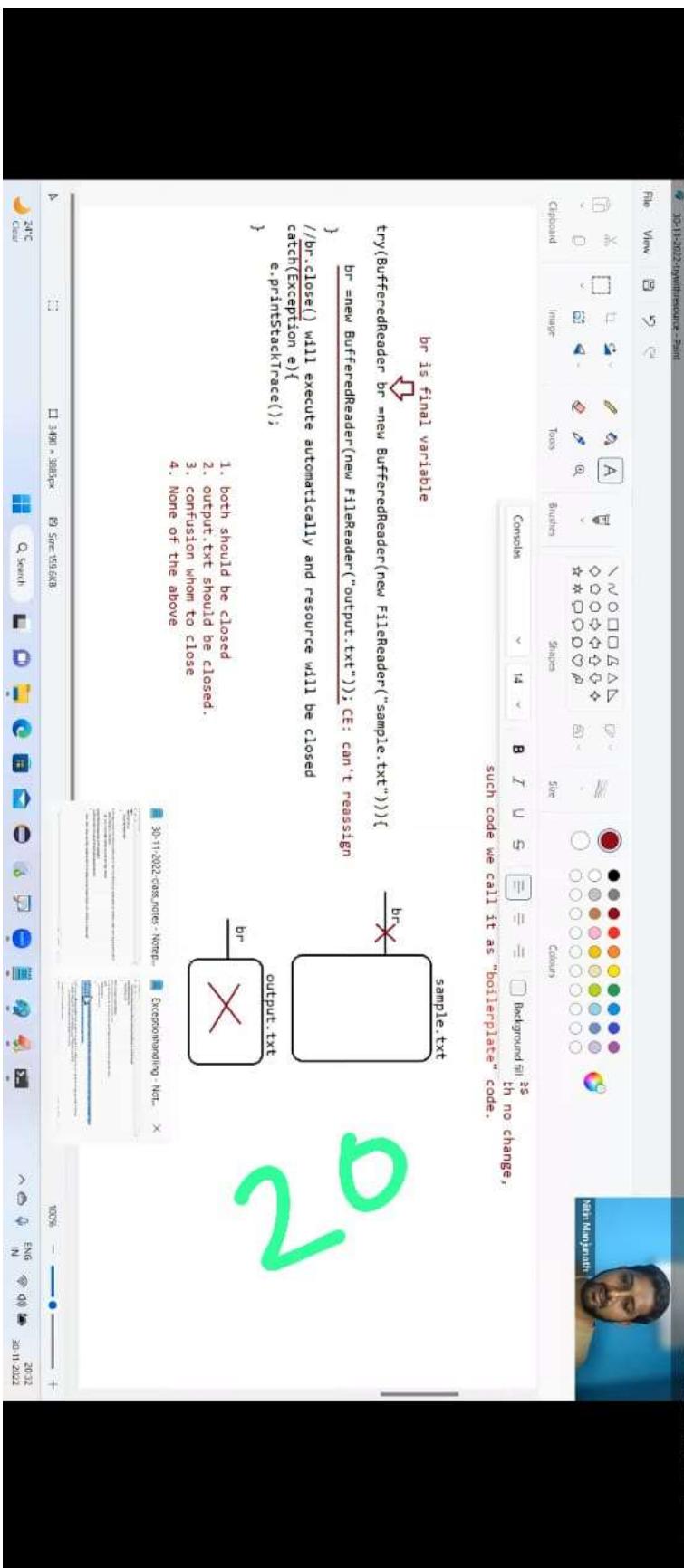
```
1
2. All resources are said to be AutoCloseable resources iff the class implements an interface called "java.lang.AutoCloseable"
either directly or indirectly
eg:: java.io package classes, java.sql.package classes

public interface java.lang.AutoCloseable {
    public abstract void close() throws java.lang.Exception;
}

Note: whichever class has implemented this interface those classes objects are referred as "resources".
1

3. All resource reference by default are treated as implicitly final and hence we can't perform reassignment within try block.
try(BufferedReader br=new BufferedReader(new FileReader("abc.txt")){
    br=new BufferedReader(new FileWriter("abc.txt"));
}
output::CE: can't reassign a value
```

19



A screenshot of a Windows terminal window titled "30.11.2022-class-primer - Notepad". The window contains the following Java code:

```
30.11.2022-class-primer - Notepad
File Edit View
try(R{
    //valid
}
4. until 1.6 version try should compulsorily be followed by either catch or finally, but from
1.7 version we can take only take try with resources without cath or finally.
5. Advantage of try with resources concept is finally block will become dummy because we are not required to close
resources explicitly.
```

The terminal window has a green hand-drawn arrow pointing from the word "finally" in the fourth bullet point to the "finally" keyword in the code. The window also shows a video call interface with a person's face and the name "Nitin Manjrekar".

```
1.6
public static void main(String[] args) {
    //JDK1.6v finally is hero
    try {
        //Exception occurs => Abnormal becoz no catch block
    } finally{
        //resource releasing logic
    }
}

vs

//JDK1.7v onwards finally is zero
try(R{
    //Exception occurs => Abnormal becoz no catch block
})
```

File Edit View Search Document Project Tools Browser Export Window Help

Directory Object Functions

(D) New Volume

D: Whopping classes

IntelliJ

```
public static void main(String[] args) {
    1
    2
    3
    4
    5
    6
    7
    8
    9
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
}
```

try(R1){
try(R2){
try(R3){
}catch (Exception e1){
}
}catch (Exception e2){
}
}
}catch (Exception e3){
}

23

Java (*.java)

For Help press F1

24C

Carrie

Int 17 col 18 27 28 PC ANSI

~ ^ < > < > ENG ◊ IN 20.11.2028

File Edit View

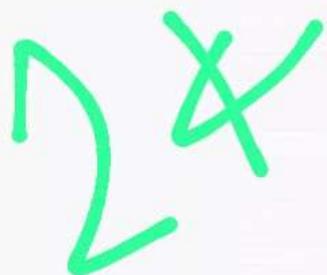
1.7 version we can take only take try with resources without catch or finally.

```
try(R){  
    //valid  
}
```

5. Advantage of try with resources concept is finally block will become dummy because we are not required to close resources explicitly.

6. try with resource nesting is also possible.

```
try(R1){  
    try(R2){  
        try(R3){  
            ...  
        }  
    }  
}
```



Ln 97 Col 1

Q. Search

24C Clear

10% Windows (CEP)

UTF-8

ENG

DE

IN

20.11.2022



25

```
Emacs - (D:\Work\classes\test.java)
File Edit View Search Document Project Tools Brower Egine Window Help
D:\ New Volume Directory Object Functions 1 2 3 4 5 6 7
D:\ Writing classes 8 9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

/*
 * smoothful execution no exception and if exception occurs also we have
 * handling logic in our application.
 */
//Abnormal termination

/*
Execution ocured and resulted in exception and if we don't have
handling logic in our application and if the exception object delegated to
JVM(defualt exception handler) then it is abnormal termination|
*/
```

Untitled - [D:\WhatsApp\classes\Untitled.java]

```
1 public static void main(String[] args) {  
2     try{  
3         //risky code  
4         ;;;;;;;;;;  
5         ;;;;;;;;;;  
6         ;;;;;;;;;;  
7         ;;;;;;;;;;  
8         ;;;;;;;;;;  
9         ;;;;;;;;;;  
10        ;;;;;;;;;;  
11        ;;;;;;;;;;  
12        new BufferedReader(new FileReader("sample.txt"));  
13    }catch (ArithmeticException e){  
14        e.printStackTrace();  
15    }catch(NullPointerException e){  
16        e.printStackTrace();  
17    }catch(ClassCastException e){  
18        e.printStackTrace();  
19    }catch(IOException e){  
20        e.printStackTrace();  
21    }  
22    }  
23}  
24}
```

Java (Java)
Java Test Java
For Help press F1
23°C
Cloud

26

Erkut - [D:\Work\java\Tutorial]

File Edit View Search Document Project Tools Browser Errors Window Help

Directory Object Functions

D:\ New Volume Wimpel classes

Test.java

```
1 import java.io.*;  
2  
3 public class Test {  
4     public static void main(String[] args) {  
5         //JDK1.7V{try with multi-catch block}  
6         try{  
7             //risky code  
8             ;  
9             ;  
10            ;  
11            ;  
12            new BufferedReader(new FileReader("sample.txt"));  
13        }catch (ArithmaticException | NullPointerException e){  
14            //handling code  
15            e.printStackTrace();  
16        }catch(ClassCastException | IOException e){  
17            //handling code  
18            e.printStackTrace();  
19        }  
20    }  
21 }  
Java (*.java)
```

For Help press F1

23°C Clear

Nitin Mehta

In 5 col 46 26 03 PC ANSI

~ ⌂ ⌄ ENG ⇧ ⇩ ⌂ ⌃ 20/11/2022



30-11-2022-class-prime - Notepad

File Edit View

Nitin Mangrulkar

```
MulticatchBlock
=====
Till jdk1.6, even though we have multiple exception having same handling code we have to write a
seperate catch block for every exceptions, it increases the length of the code and reviews
readability.

try{
    ...
    ...
    ...
    ...
}

}catch(ArithmeticException ae){
    ae.printStackTrace();
}
}catch(NullPointerException ne){
    ne.printStackTrace();
}
}catch(ClassCastException ce){
    System.out.println(ce.getMessage());
}
}catch(IOException ie){
    System.out.println(ie.getMessage());
}
```

20

```
30-11-2022-class-prime - Notepad
File Edit View
logic
=====
try{
...
...
...
...
}

}catch(ArithmeticException ae){
    ae.printStackTrace();
}catch(NullPointerException ne){
    ne.printStackTrace();
}catch(ClassCastException ce){
    System.out.println(ce.getMessage());
}catch(IOException ie){
    System.out.println(ie.getMessage());
}


```

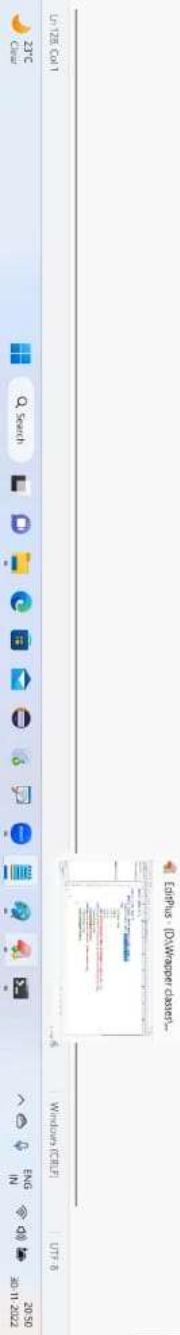
29

```
30.11.2022-class_name - Notepad  
File Edit View  
System.out.println(ce.getMessage());  
}catch(IOException ie){  
    System.out.println(ie.getMessage());  
}
```

To overcome this problem SUNMS has introduced "Multi catch block" concept in 1.7 version

```
try{  
    ...  
    ...  
    ...  
    ...  
}  
}catch(ArithmaticException | NullPointerException e){  
    e.printStackTrace();  
}  
}catch(ClassCastException | IOException e){  
    e.printStackTrace();  
}
```

30



File Edit View Search Document Project Icons Browser Export Window Help

Directory Object Functions

(D) New Volume D:\ WinRAR classes

Test.java

```
1 import java.io.*;  
2  
3 public class Test {  
4     public static void main(String[] args) {  
5         //JDK1.7V(try with multi-catch block)  
6         try{  
7             //risky code  
8             ;;;;  
9             ;;;;  
10            ;;;;  
11            ;;;;  
12            int a =10/0;  
13            System.out.println(a);  
14        }catch (ArithmaticException | Exception e){  
15            //handling code  
16            e.printStackTrace();  
17        }  
18    }  
19  
20 }  
21 }
```

Java (*.java)

For Help press F1

23°C Clear

File Edit View Search Document Project Icons Browser Export Window Help

Directory Object Functions

(D) New Volume D:\ WinRAR classes

Test.java

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21 }
```

In 12 col 25 27 00 PC ANSI

ENGLISH QWERTY IN 2011-2022



23 TC

Q Search

Minimise Maximize Close

Minimise Maximize Close

Nitin Manjrekar

20/11/2022

ENG ⌂ ⌃ ⌁ ⌂ ⌃ ⌁

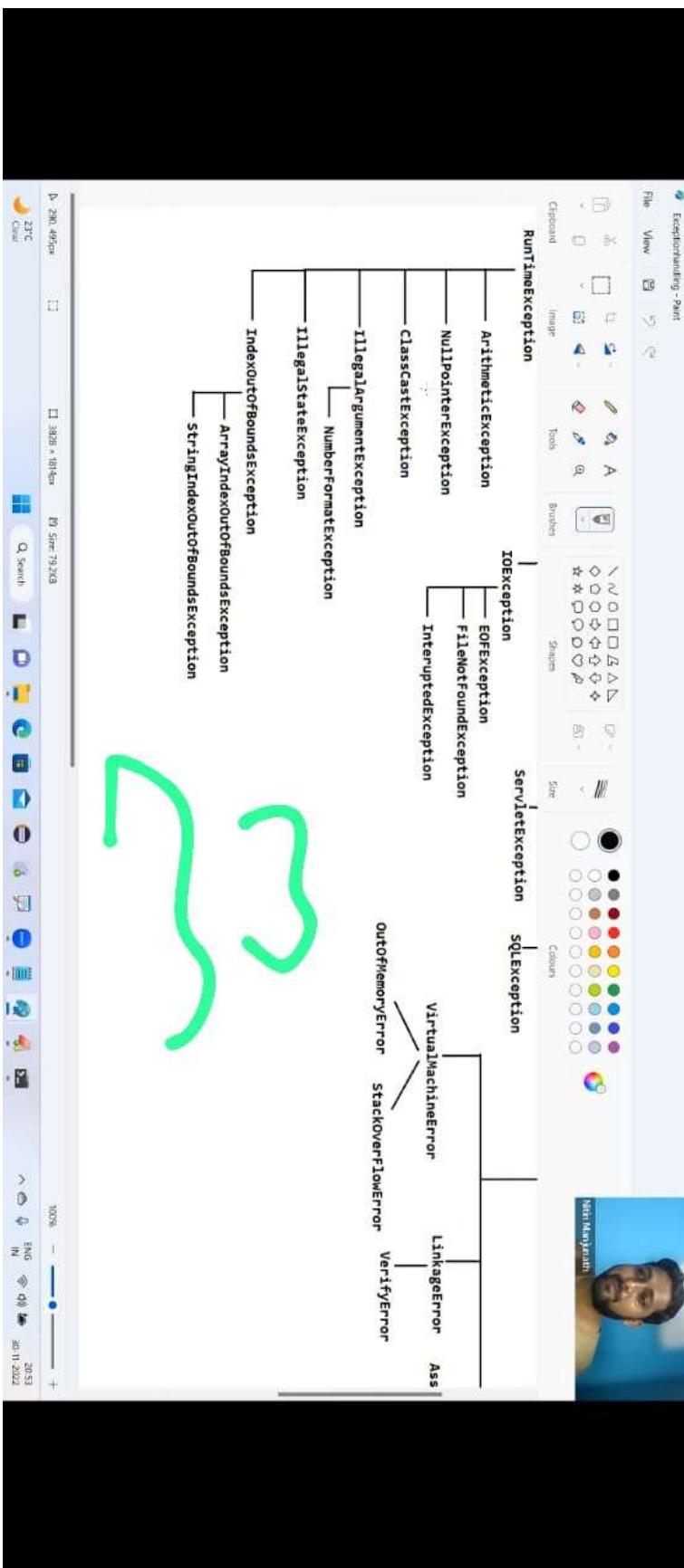
Command Prompt

```
D:\Wrapper\classes>javac Test.java
D:\Wrapper\classes>javac Test.java
Test.java:12: error: ';' expected
        int a =10/0
               ^
1 error

D:\Wrapper\classes>javac Test.java
Test.java:15: error: Alternatives in a multi-catch statement cannot be related by subclassing
    }catch (ArithmetricException | Exception e){
                           ^
Alternative ArithmetricException is a subclass of alternative Exception
1 error

D:\Wrapper\classes>
```

12



File Edit View Search Document Project Tools Browser Egret Window Help

Directory Object Functions

[D] New Volume D: Wampack classes

Test.java

```
1 import java.io.*;
2
3 public class Test {
4     public static void main(String[] args) {
5         //JDK1.7V(try with multi-catch block)
6         try{
7             //risky code
8             ;;;;;;;;;;
9             ;;;;;;;;;;
10            ;;;;;;;;;;
11            ;;;;;;;;;;
12            int a = 10/0;
13            System.out.println(a);
14            new BufferedReader(new FileReader("sample.txt"));
15            Class.forName(args[0]);
16
17        }catch (ArithmaticException | NullPointerException | IOException | ClassNotFoundException
18             //handling code
19             e.printStackTrace();
20
21
22 }
```

Java (*.java)

For Help press F1

23°C Clear

Nitin Mangat

34

23°C
Cloudy

Q Search

Min Max Min Max

^ ⌂ ⌄ ENG ⌂ ⌄ IN 20:56
20.11.2022

```
Command Prompt

D:\Wrapper\classes>javac Test.java
D:\Wrapper\classes>javac Test.java
Test.java:17: error: exception ClassNotFoundException is never thrown in body of corresponding try statement
        }catch (ArithmaticException | NullPointerException | IOException | ClassNotFoundException e){
                           ^
1 error

D:\Wrapper\classes>javac Test.java
D:\Wrapper\classes>
```

30-11-2022-class-prime - Notepad

File Edit View

Exception Handling

- 1. try with resource.
- 2. try with multicatch block.
- 3. Rules of Overriding associated with Exception.

Remaining topics to be discussed

- 1. instanceof vs isInstanceOf(Object obj)
- 2. How to create a userdefined package and in realtime project how it is used?

1.7 version Enhancements

- 1. try with resource
- 2. try with multicatch block

until jdk1.6, it is compulsorily required to write finally block to close all the resources which are open as a part of try block.

Un 10 Corrig 23°C Clear 100% Windows (CE/F) UTT-3 ENG ENG 20 11 2022



Rules of using try with resource

1. we can declare any no of resources, but all these resources should be seperated with ;

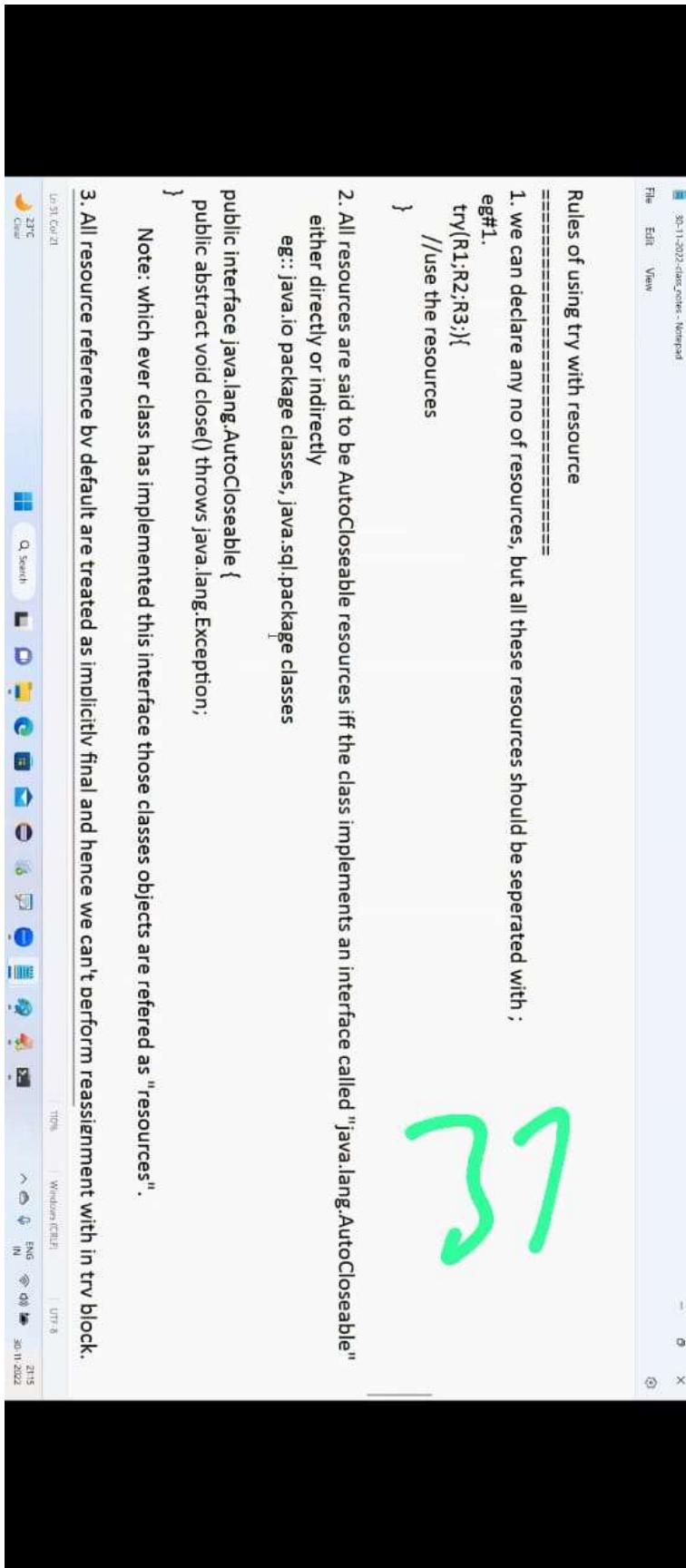
```
try(R1;R2;R3;){  
  //use the resources  
}
```

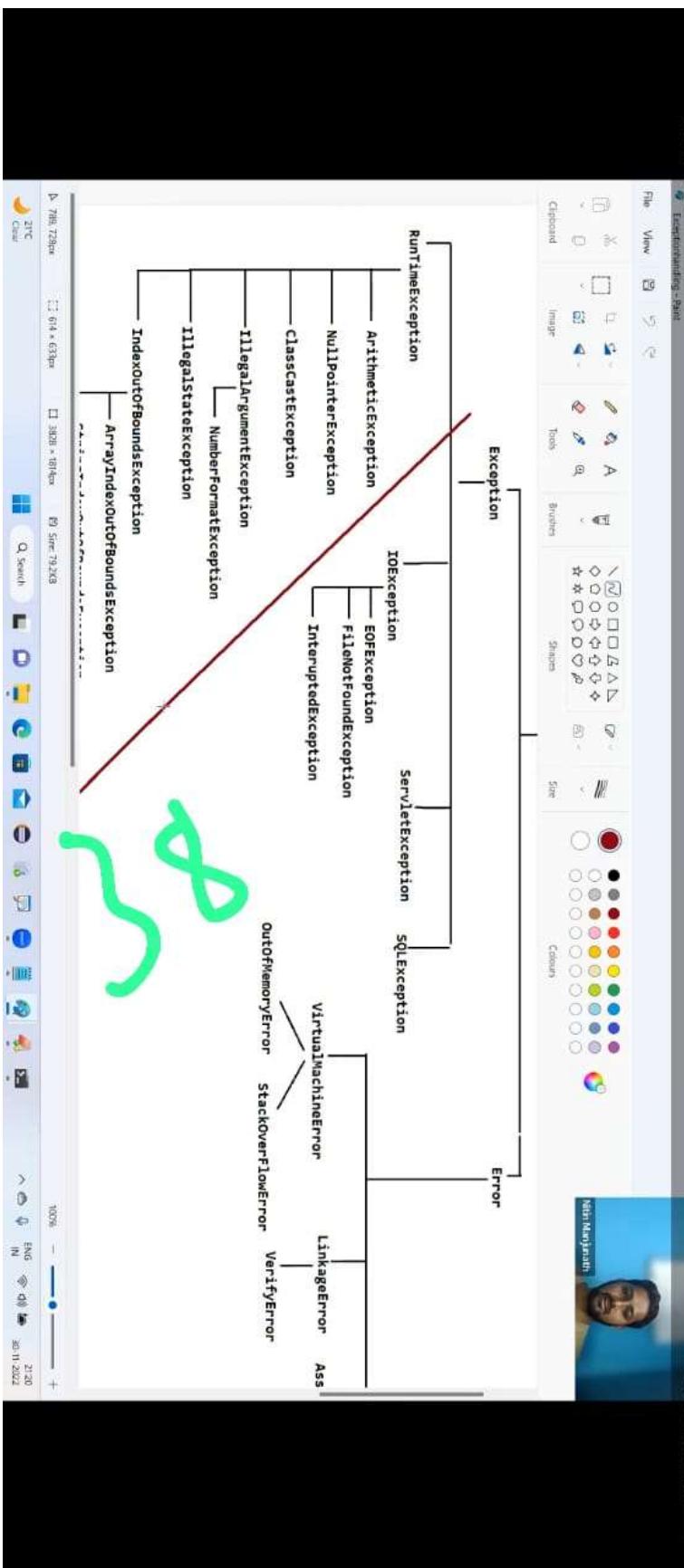
2. All resources are said to be AutoCloseable resources iff the class implements an interface called "java.lang.AutoCloseable"

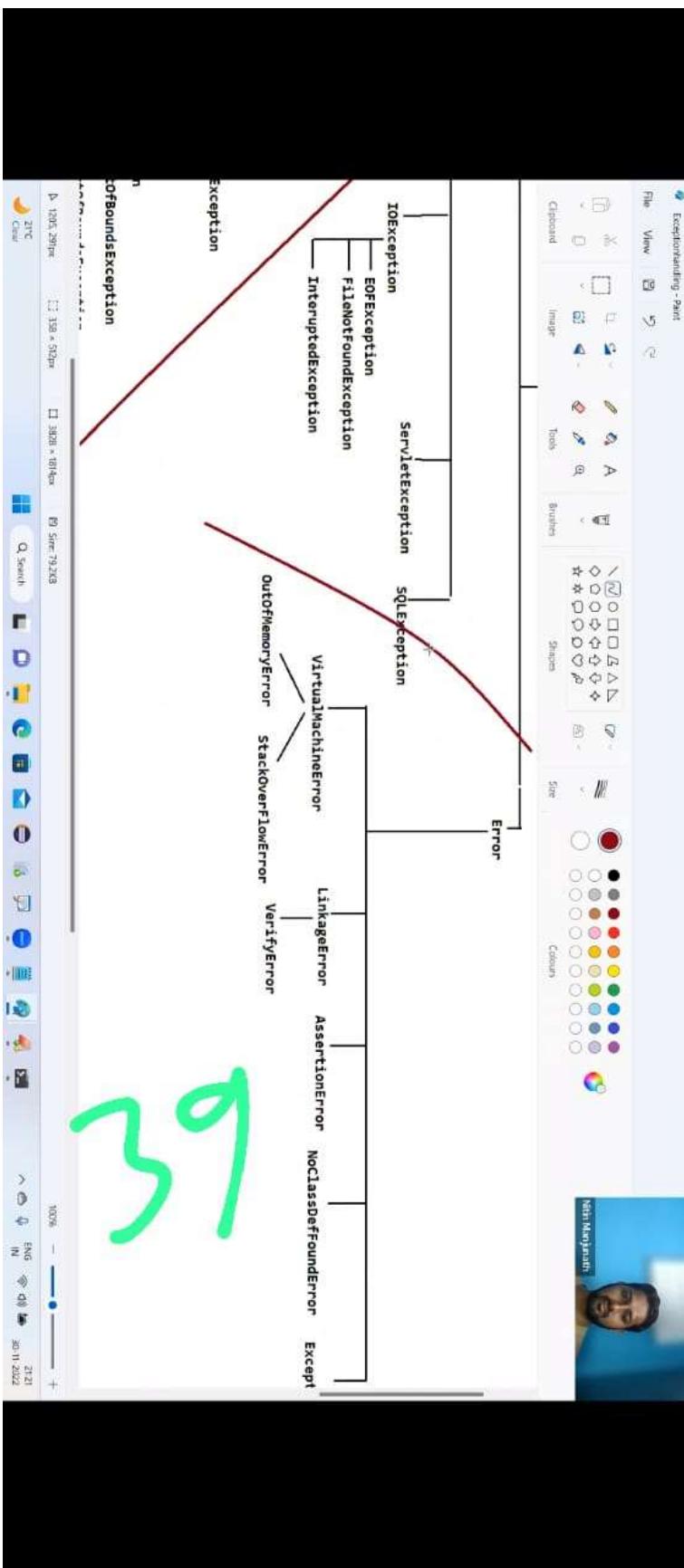
```
eg.: java.io package classes, java.sql.package classes  
public interface java.lang.AutoCloseable {  
    public abstract void close() throws java.lang.Exception;
```

Note: whichever class has implemented this interface those classes objects are referred as "resources".

3. All resource reference by default are treated as implicitly final and hence we can't perform reassignment within try block.







throw => handle the exception using catch block and throw it back the exception object to the caller.
throws => method signature and commonly used if the exception is "CheckedException".

CheckedException => compiler will check for the handling code only then compilation is successful.
eg: IOException, SQLException,.... are all checked exceptions.

UnCheckedException => compiler will not check for the handling code, but JVM will come into picture
and possibility of "successful" or "abnormal" termination.
eg: RuntimeException and its child classes
Error and its child classes are all "UncheckedException".

1



```
File Edit - [D:\Wagner classpath]\javaw.exe
File Edit New Search Document Project
New Open Save Print Window Help
D:\ New Volume D:\ Wrapper classes
Directory Clipper Functions
1 import java.io.*;
2 class Parent {
3     public void m1()throws Exception{
4         }
5     }
6 }
7 }
8 }
9 class Child extends Parent {
10    }
11
12 @Override
13 public void m1()throws IOException{
14     new BufferedReader(new FileReader("sample.txt"));
15 }
16 }
17
18 public class Test {
19     public static void main(String[] args) {
20         }
21     }
22 }
```

File Edit View

Rules of Overriding when exception is involved

While Overriding if the child class method throws any checked exception compulsorily the parent class method should throw th

Compile Time Error.

```
class Parent{
```

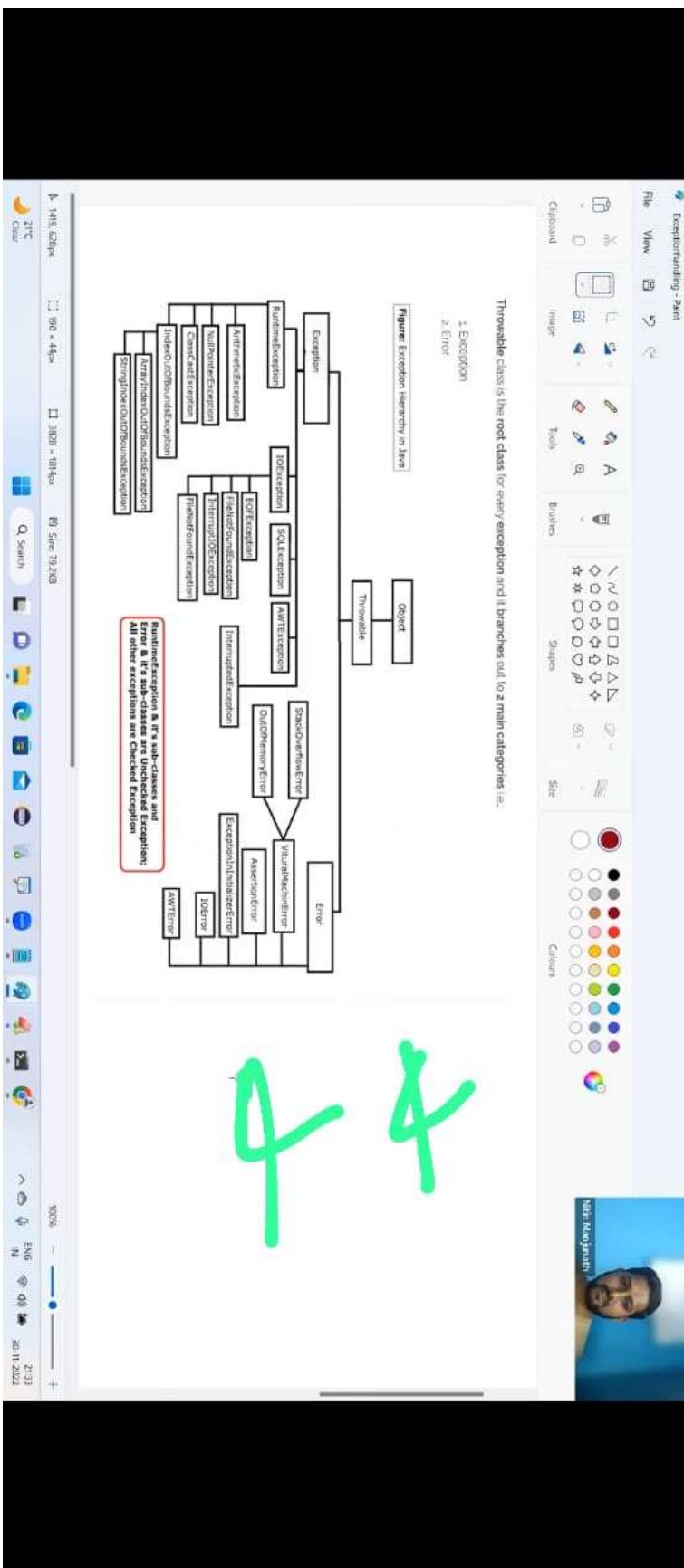
```
class Child extends Parent{  
    public void methodOne() throws Exception{}  
}  
  
error: methodOne() in Child cannot override methodOne() in Parent  
public void methodOne() throws Exception{}  
overridden method does not throw Exception
```

42



```
File Edit View  
Rules w.r.t Overriding  
=====  
parent: public void methodOne() throws Exception{}  
child : public void methodOne()  
output: valid  
  
parent: public void methodOne(){}  
child : public void methodOne() throws Exception{}  
output: invalid  
  
parent: public void methodOne()throws Exception{}  
child : public void methodOne()throws Exception{}  
  
parent: public void methodOne()throws IOException{}  
child : public void methodOne()throws IOException{}
```

43



30.11.2022-class_primer - Notepad

File Edit View

parent: public void methodOne()throws IOException{}
child : public void methodOne()throws IOException{}
output: valid

parent: public void methodOne()throws IOException{}
child : public void methodOne()throws FileNotFoundException,EOFException{}
output: valid

parent: public void methodOne()throws IOException{}
child : public void methodOne()throws FileNotFoundException,InterruptedException{}
output: invalid

parent: public void methodOne()throws IOException{}
child : public void methodOne()throws FileNotFoundException,ArithmaticException{}

4/5

10% Windows (CEP) UTF-8 ↻ ENG ⇄ DE 21:33 30.11.2022

46

```
30.11.2022 class-prime - Notepad  
File Edit View  
child : public void methodOne()throws FileNotFoundException,InterruptedException{}  
output: invalid  
  
parent: public void methodOne()throws IOException{}  
child : public void methodOne()throws FileNotFoundException,ArithmaticException{}  
output: valid  
  
parent: public void methodOne()  
child : public void methodOne()throws ArithmaticException,NullPointerException,RuntimeException{}  
output: valid
```

File Edit View 30-11-2012<notes>

```
parent: public void methodOne()  
child : public void methodOne() throws ArithmeticException,NullPointerException,RuntimeException{}  
output: valid
```

```
parent: public void methodOne()throws IOException{}  
child : public void methodOne()throws Exception{}  
output: invalid
```

```
parent: public void methodOne(){throws Throwable{}  
child : public void methodOne(){throws IOException{}}  
output: valid
```

47



ITALICUS

1. We can use the `instanceof` operator to check whether the given an object is particular type or not.

X => class/interfaceName

८७

```
ArrayList al =new ArrayList();
al.add(new Student());
al.add(new Cricketer());
al.add(new Customer());
```

48



```
30.11.2022-class-prime - Notepad  
File Edit View  
a.add(new Student());//0th position  
a.add(new Cricketer());//1st position  
a.add(new Customer());//2nd position  
  
Object o=a.get(0);  
if( o instanceof Student){  
    Student s =(Student)o;  
    //perform operation on student  
}else if (o instanceof Cricketer){  
    Cricketer c =(Cricketer)o;  
    //perform operation on student  
}else if(o instanceof Customer){  
    Customer c =(Customer)o;  
    //perform operation on customer  
}
```

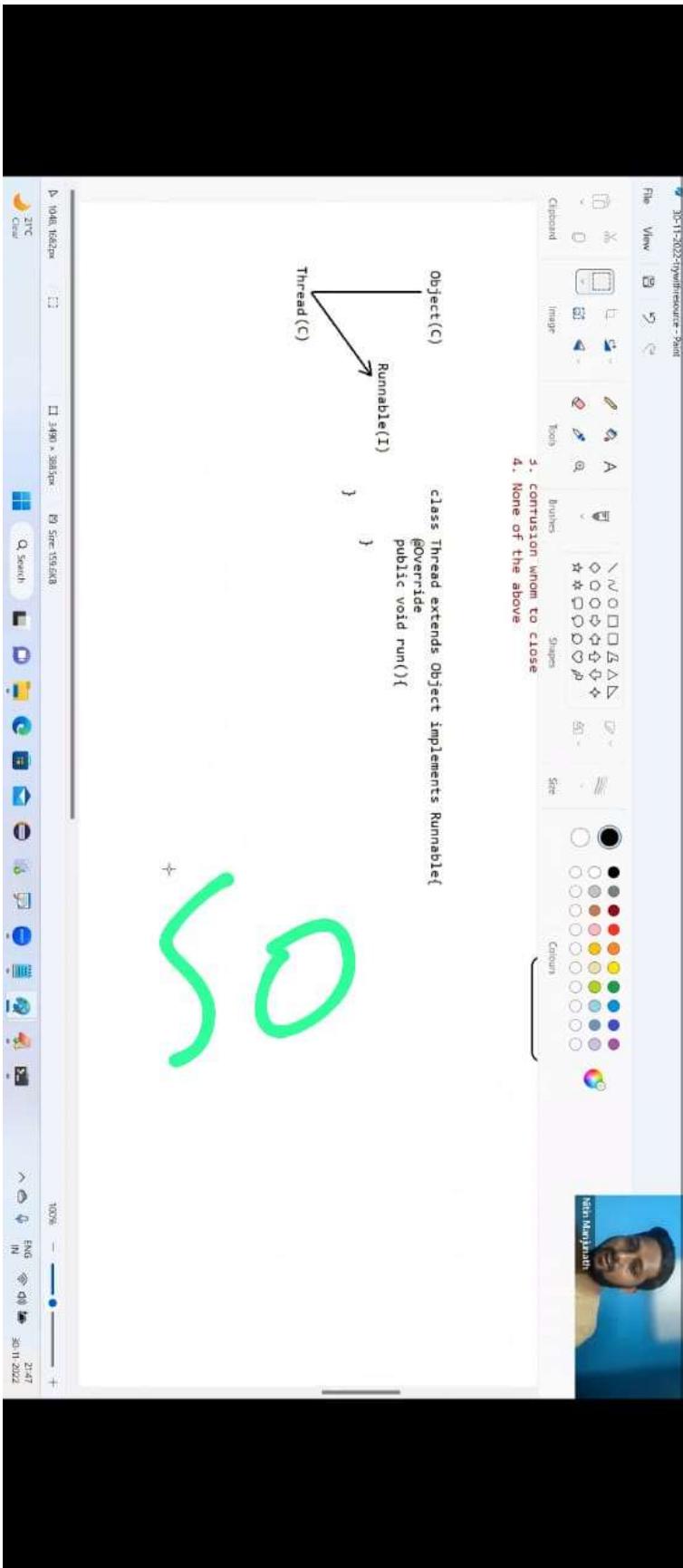
49

20°C
Cloudy



Q Search





The screenshot shows a Java IDE interface with the following details:

- Title Bar:** "EduBou - [D:\Wamp\www\classmate\src]
- Toolbar:** Standard file operations like New, Open, Save, Print, etc.
- Menu Bar:** File, Edit, View, Search, Document, Project, Tools, Browser, Editor, Window, Help.
- Toolbars:** Directories, Clipboard, Functions.
- Status Bar:** In 11 cd 73 19 00 PC ANSI 21:50 30/11/2022
- Code Area:** Shows a Java code snippet with annotations and a large green 'S' drawn over it.

```
1 import java.io.*;  
2  
3  
4  
5 //  
6 // public class Test {  
7 //     public static void main(String[] args) {  
8 //         Thread t = new Thread();  
9 //         System.out.println(t instanceof Object); //true  
10 //         System.out.println(t instanceof Thread); //true  
11 //         System.out.println(t instanceof Runnable); //true  
12 //         System.out.println(t instanceof String); //CE|  
13 }  
14  
15  
16  
17  
18  
19
```
- Bottom Right:** A small portrait of a person named Nitin Naipatni.


```
30-11-2022.class.prime - Notepad  
File Edit View  
  
eg: String s= new String("sachin");  
System.out.println(s instanceof Thread);//CE  
  
Thread t=new Thread();  
System.out.println(t instanceof String);//CE  
  
=> Whenever we are checking the parent object is child type or not by using instanceof operator that we get false.  
Object o=new Object();  
System.out.println(o instanceof String );//false  
  
Object o=new String("ashok");  
System.out.println(o instanceof String); //true  
  
=> For any class or interface X null instanceof X is always returns false  
System.out.println(null instanceof X); //false
```

S 3

49.1-2022-class_review - Notepad

File Edit View

=> For any class or interface X null instanceof X is always returns false

```
System.out.println(null instanceof X); //false
```

```
public class Test {  
    public static void main(String[] args) {  
        Object t = new Thread();  
        System.out.println(t instanceof Object); //true  
        System.out.println(t instanceof Thread); //true  
        System.out.println(t instanceof Runnable); //true  
        System.out.println(t instanceof String); //false  
        System.out.println(t instanceof Object); //false  
    }  
}
```

SA

11/27/2022 2:22:01 PM

Windows (CEP)

UTF-8

20C

Clear

Q. Search

File Edit View

Difference between instanceof and isInstance():

instanceof

instance of an operator which can be used to check whether the given object is particular type or not. We know at the type at beginning it is available.

```
eg: String s = new String("sachin");
System.out.println(s instanceof Object); //true
//if we know the type at the beginning only.
```

`isinstance()` is a method , present in class `Class` , we can use `isinstance()` method to checked whether the given object is particular type or not . We don't know at the type at beginning it is available Dynamically at Runtime.



File Edit View

//if we know the type at the beginning only.

`isinstance()` is a method, present in class `Class`, we can use `isinstance()` method to checked whether the given object is particular type or not. We don't know at the type at beginning it is available Dynamically at Runtime.

```
class Test{  
    }  
    public static void main(String[] args) {  
        Test t = new Test();  
        System.out.println(Class.forName(args[0]).isInstance(t));//arg[0] --- We don't know the type at beginning  
    }  
}  
  
java Test Test //true  
java Test String //false  
java Test Object //true
```

56

File Edit View

Q>

Given:

```
abstract public class Employee {  
    protected abstract double getSalesAmount();  
    public double getCommission() {  
        return getSalesAmount() * 0.15;  
    }  
}  
class Sales extends Employee {  
    17. // insert method here  
}
```

Which two methods, inserted independently at line 17, correctly complete the Sales class? (Choose two.)

- A. double getSalesAmount() { return 1230.45; }
- B. public double getSalesAmount() { return 1230.45; }
- C. private double getSalesAmount() { return 1230.45; }
- D. protected double getSalesAmount() { return 1230.45; }

Answer: BD



*30-11-2022_mypage_session - Microsoft Edge
File Edit View

What is the result?
1. public class A{

- ```
5. public static int getInstanceCount() {
6. return counter;
7. }
8.
9. public A(){
10. counter++;
11. }
12.
13.}
```

59



30-11-2022\_294ppc\_session - Notepad

File Edit View

```
2.
3. private int counter = 0;
4.
5. public static int getInstanceCount() {
6. return counter;
7. }
8.
9. public A() {
10. counter++;
11. }
12.
13.}
```

A. Compilation of class A fails.  
B. Line 28 prints the value 3 to System.out.  
C. Line 28 prints the value 1 to System.out.  
D. A runtime error occurs when line 25 executes.  
E. Compilation fails because of an error on line 28.

answer: A

60

Un 30 Col 1 20°C Clear

Top Windows (CEP) UTF-8 ↻ ENG ⇄ DE 🇩🇪 30-11-2022

Given:

```
1. public class A {
2. public void doit() {
3. }
4. public String doit() {
5. return "a";
6. }
7. public double doit(int x) {
8. return 1.0;
9. }
10.}
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails because of an error in line 7.
- C. Compilation fails because of an error in line 4.
- D. Compilation succeeds and no runtime errors with class A occur.

Answer: C

61

```

class Inner {
 private int x;
 public void setx(int x){ this.x = x; }
 public int getx(){ return x; }
}

Outer o = new Outer();
Inner i = new Inner();
int n = 10;
i.setx(n);
o.sety(1);

answer: produce the output 100?

// insert code here 29
System.out.println(o.gety() - getx());

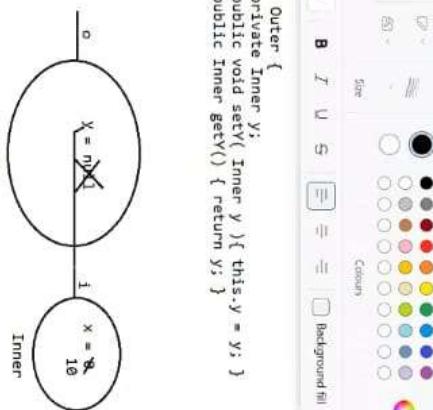
```

o  
X = n  
Outer

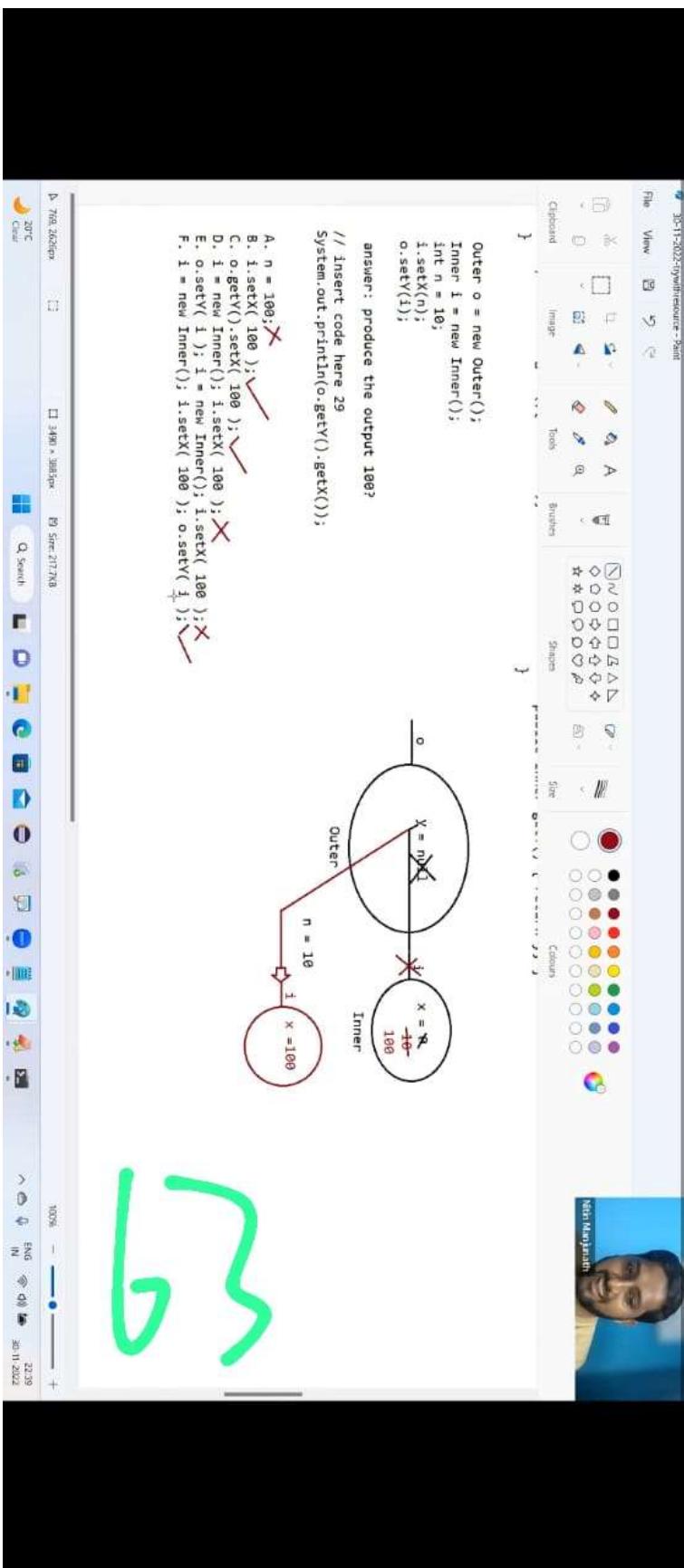
n = 10

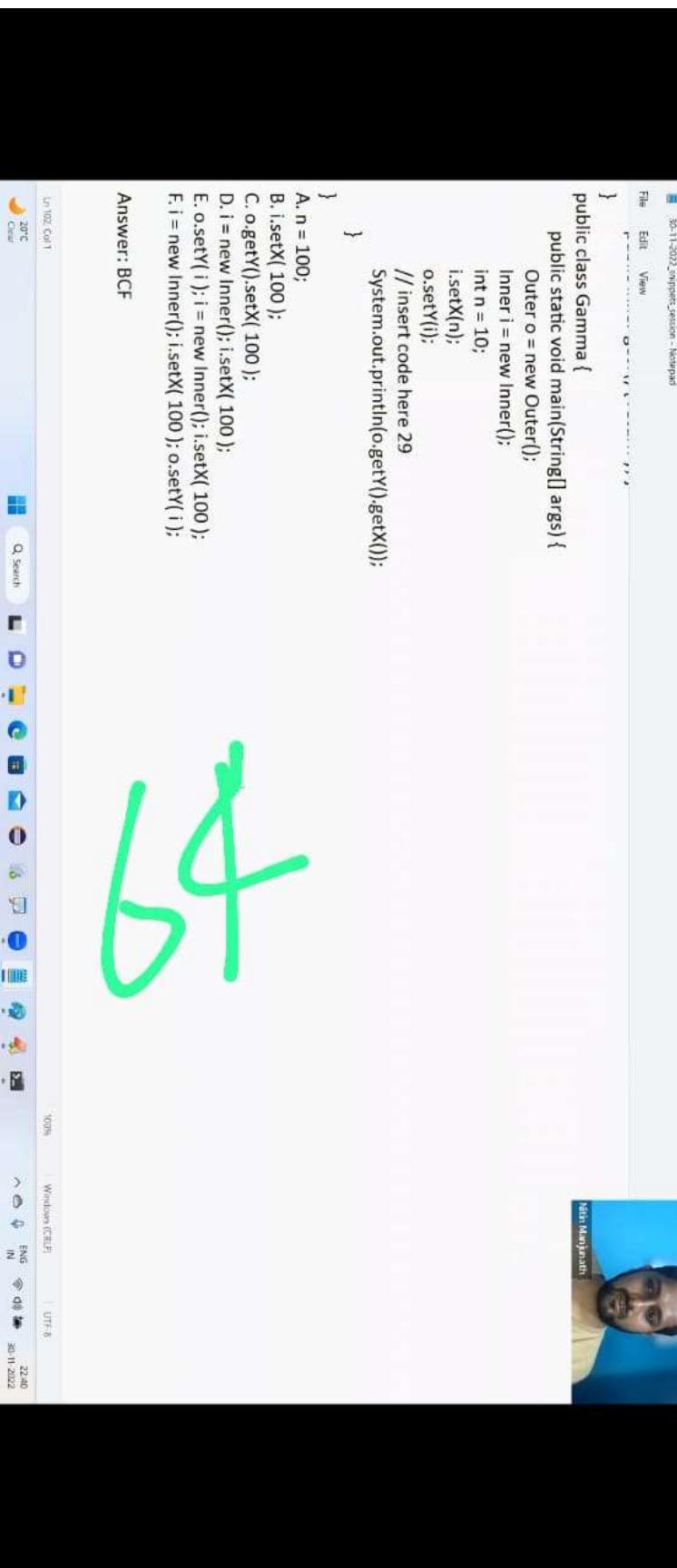
i  
X = 10  
Inner

B. i.setx( 100 );
C. o.gety().setx( 100 );
D. i = new Inner(); i.setx( 100 );
E. o.sety( 1 ); i = new Inner();
F. i = new Inner(); i.setx( 100 );
G. o.sety( 1 );



L2





```
30.11.2022_29inputs_session - Notepad
File Edit View
p
}
public class Gamma {
 public static void main(String[] args) {
 Outer o = new Outer();
 Inner i = new Inner();
 int n = 10;
 i.setX(n);
 o.setY(i);
 // insert code here 29
 System.out.println(o.getY(), getX());
 }
}
A. n = 100;
B. i.setX(100);
C. o.getY(),setX(100);
D. i = new Inner(); i.setX(100);
E. o.setY(i); i = new Inner(); i.setX(100);
F. i = new Inner(); i.setX(100); o.setY(i);
```

64

Answer: BCF

```
*192.168.2.227_2022-09-25_19-45-45 - Notepad
File Edit View
public class Base {
 public static final String FOO = "foo";
 public static void main(String[] args) {
 Base b = new Base();
 Sub s = new Sub();

 System.out.print(Base.FOO); //foo
 System.out.print(Sub.FOO); //bar
 System.out.print(b.FOO);
 System.out.print(s.FOO);
 System.out.print(((Base) s).FOO);
 }
}
class Sub extends Base {
 public static final String FOO = "bar";
}
What is the result?
A. foofoofoofoo
B. foobarfoobarbar
C. foobarfoofoofoo
D. foobarfoobarbarfoo
E. barbarbarbarbar
```

65



Clear



Q Search



D



E



F



G



H



I



J



K



L



M



N



O



P



Q



R



S



T

100%  
110.0x33

Windows (CEP)

UTF-8

~

^

≡

ENG

IN

2011-2243

2011-2243



66

```
File Edit View
System.out.print(Base.FOO);//foo
System.out.print(Sub.FOO);//bar
System.out.print(b.FOO);//foo
System.out.print(s.FOO);//bar
System.out.print(((Base) s).FOO);//foo
}
class Sub extends Base{
 public static final String FOO = "bar";
}
What is the result?
A. foofoofoofoo
B. foobarfoobarbar
C. foobarfoofoofoo
D. foobarfoobarfoo
E. barbarbarbarbar
F. foofoofoobarbar
G. foofoofoobarfoo
```

Answer: D

\*IP-31-2022\_Simonee\_Desson - Notepad

File Edit View

Given:

```
1. class Mammal {
2.
3.
4. class Raccoon extends Mammal { //Raccoon IS-A Mammal, Raccoon HAS-A mama|
5. Mammal m = new Mammal();
6.
7.
8. class BabyRaccoon extends Mammal { //BabyRaccoon IS-A Mammal|
9.
10.
```

Which four statements are true? (Choose four.)

- A. Raccoon is-a Mammal.
- B. Raccoon has-a Mammal.
- C. BabyRaccoon is-a Mammal.
- D. BabyRaccoon is-a Raccoon.
- E. BabyRaccoon has-a Mammal.
- F. BabyRaccoon is-a BabyRaccoon.

Answer: ABCE

67

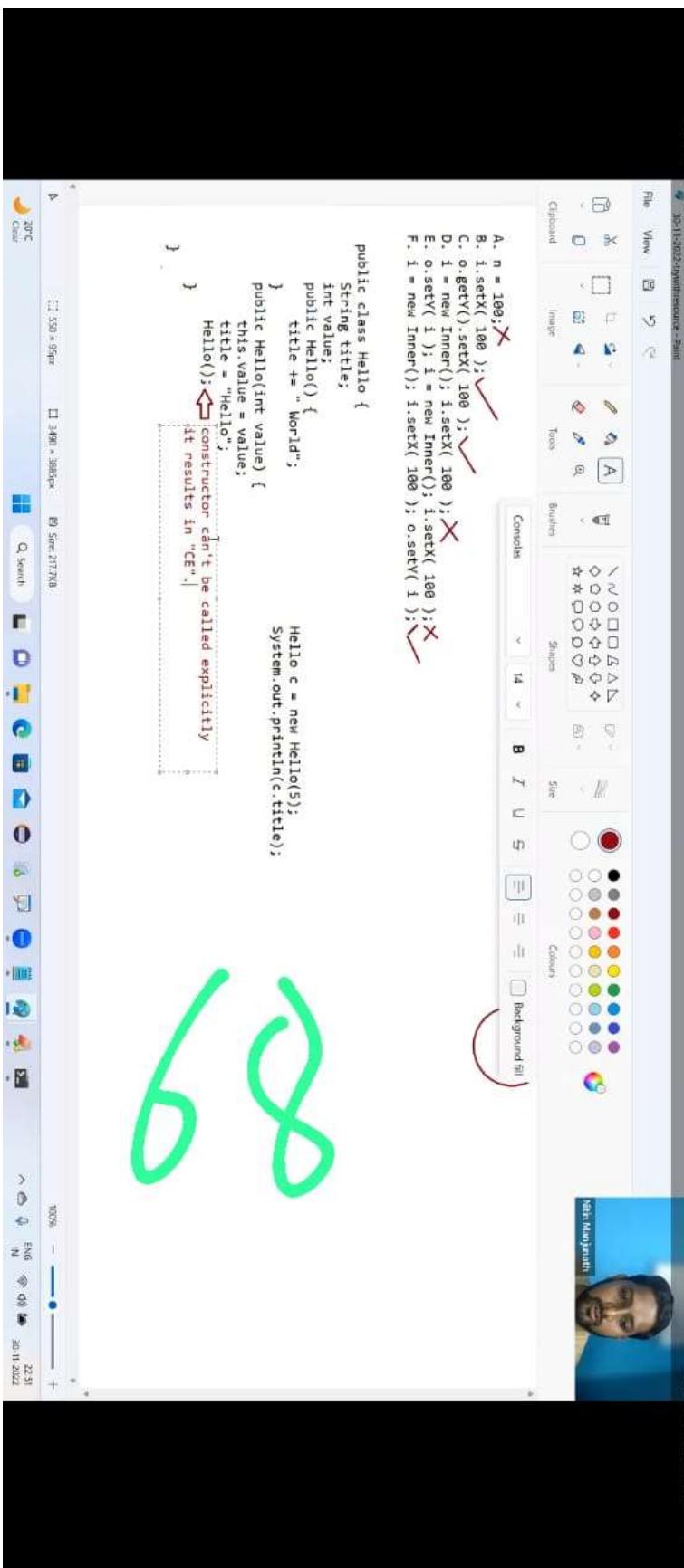


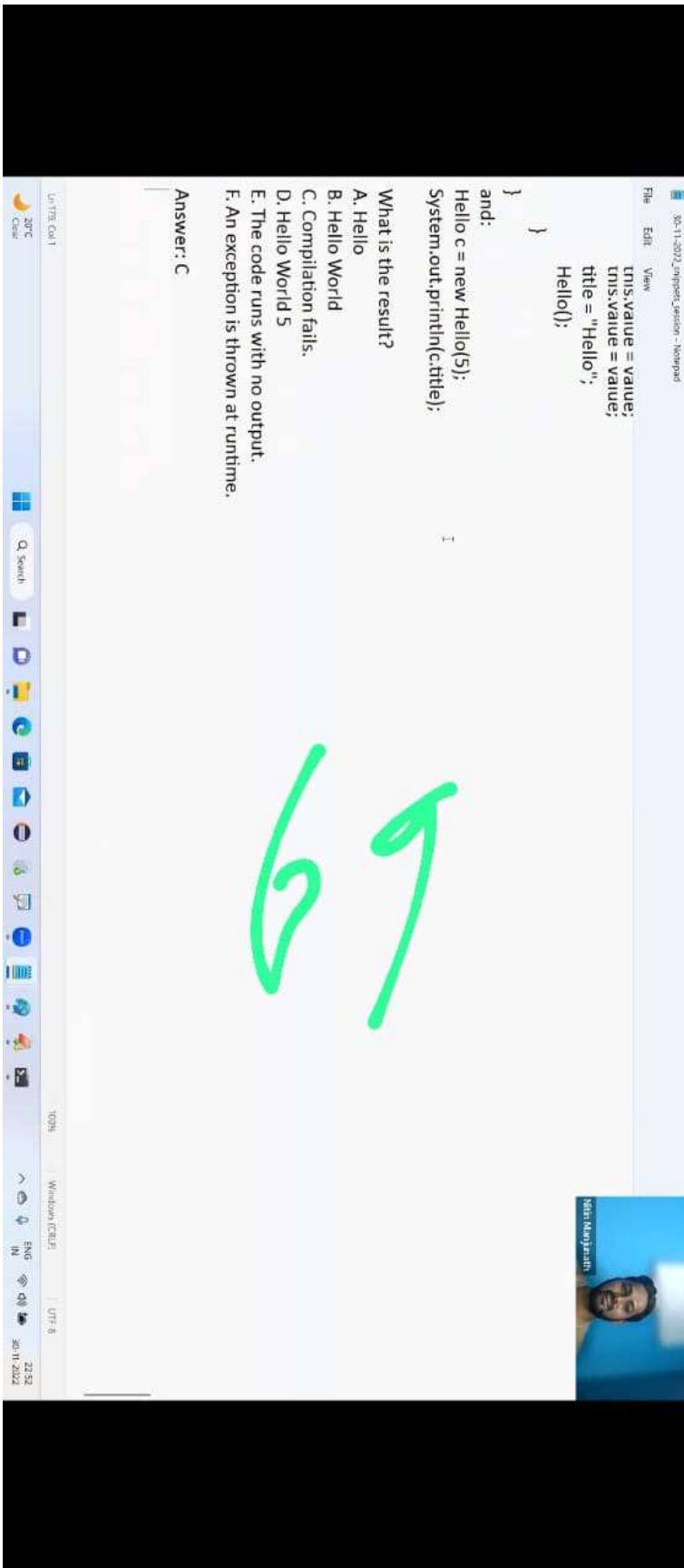
Niran Manjunath

Un 1986, CoEd 63

20°C Clear

TONK Windows (C:\EF7) UTF-8 ENG ⌂ IN 20.11.2022 22:47





30.11.2022\_294ppcSession - Notepad

File Edit View

```
this.value = value;
this.value = value;
title = "Hello";
Hello();
```

}

and:

```
Hello c = new Hello(5);
System.out.println(c.title);
```

What is the result?

- A. Hello
- B. Hello World
- C. Compilation fails.
- D. Hello World 5
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: C

69

1778 Col 1

20°C Clear

Q. Search

Windows [CE/F] | UTF-8

^ ⌂ ⌄ ENG ⌂ ⌄ IN 22:52 30.11.2022