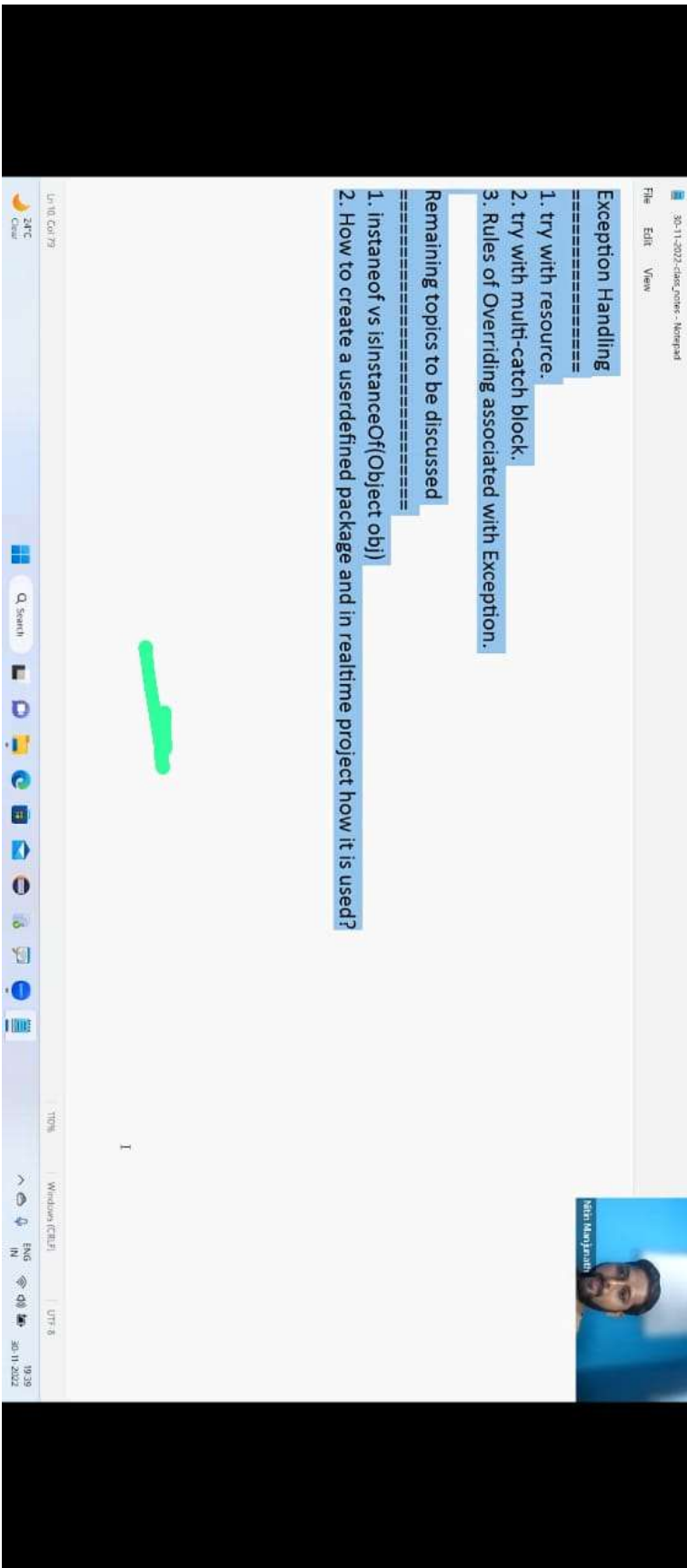


# Java Exception Part 4

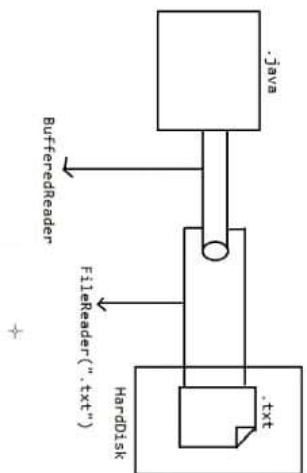


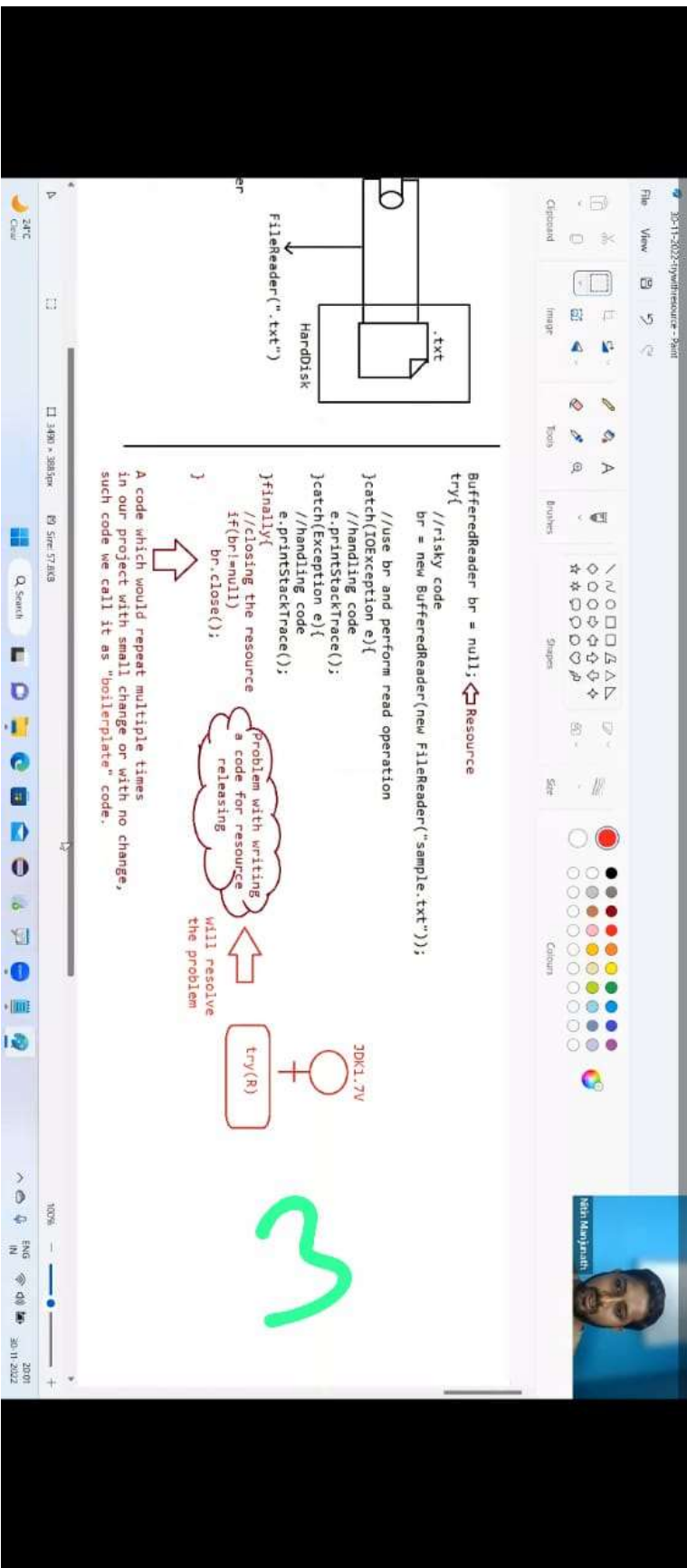
The screenshot shows a Notepad++ window titled "30-11-2022-classes notes - Notepad++". The text inside the window is as follows:

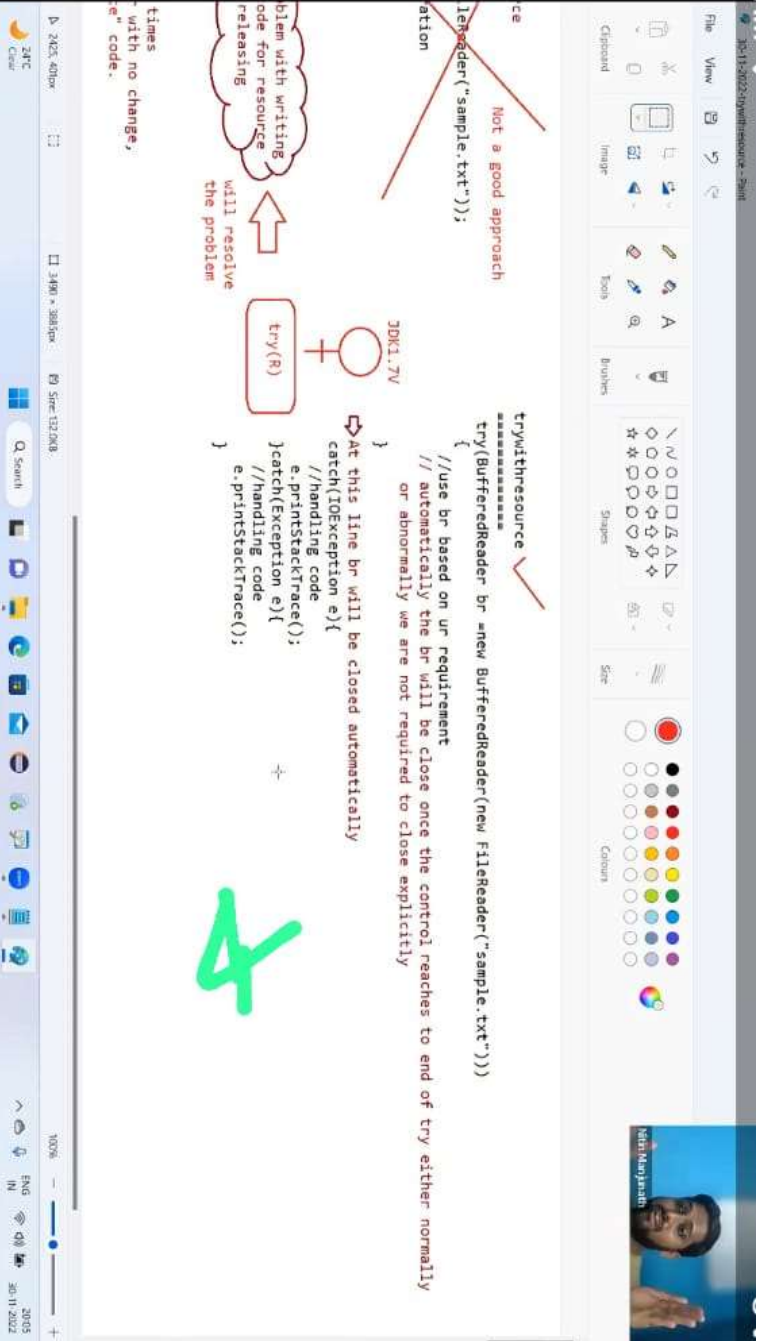
```
Exception Handling
=====
1. try with resource.
2. try with multi-catch block.
3. Rules of Overriding associated with Exception.

Remaining topics to be discussed
=====
1. instanceof vs instanceof(Object obj)
2. How to create a userdefined package and in realtime project how it is used?
```

A green rectangular highlight is visible in the center of the window. In the bottom right corner, there is a small video feed of a man with a beard, identified by the name "Nani Rajarath" below it. The Windows taskbar is visible at the bottom, showing the search bar, task view button, and several application icons. The system tray on the right shows the date and time as "20-11-2022 19:39".







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

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();  
    }  
}
```

Handwritten green signature

Problems in the apporach



```
}  
}
```

### 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){  
    ..  
}
```

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
}
```



Command Prompt

X + V

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>

8



Test (sachin)

28°C  
Clear



Search



## 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>java java.lang.String  
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  
Compiled from "String.java"  
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(int, int, int);  
 public java.lang.String(byte[], int, int, int);  
 public java.lang.String(byte[], int, int);  
 public java.lang.String(String, int, int, java.lang.String) throws java.io.UnsupportedEncodingException;  
 public java.lang.String(byte[], int, int, java.nio.charset.Charset);  
 public java.lang.String(String, java.nio.charset.Charset);  
 public java.lang.String(byte[], int, int);  
 public java.lang.String(String, java.lang.StringBuffer);  
 public java.lang.String(String, java.lang.StringBuilder);  
 public java.lang.String(char[], boolean);  
 public int length();  
 public boolean isEmpty();  
 public char charAt(int);  
 public int codePointBefore(int);  
 public int codePointBefore(int, int);  
 public int offsetByCodePoints(int, int);

6

0.02

Command Prompt

X

+

v

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 int read() throws java.io.IOException;  
    public int read(char[], int, int) throws java.io.IOException;  
    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();  
    static {};  
}
```

53 D:\Wrapper classes>

24°C  
Clear

Windows

Q Search

File Explorer

Google Chrome

Microsoft Edge

Visual Studio Code

PowerShell

Terminal

Task View

System

Network

Storage

Security

Windows Defender

Windows Update

Windows Firewall

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

Windows Defender

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;
```

Command Prompt

```
D:\Wrapper\classes>javap java.sql.Statement
Compiled from "Statement.java"
```

```
public interface 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.SQLException 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;
```

Nishu Marjan



21

24°C  
Clear



Q Search



```
D:\Wrapper\classes>javap java.lang.Integer
Compiled from "Integer.java"
public final class java.lang.Integer extends java.lang.Number implements java.lang.Comparable<java.lang.Integer> {
    public static final int MIN_VALUE;
    public static final int MAX_VALUE;
    public static final java.lang.Class<java.lang.Integer> TYPE;
    static final char[] digits;
    static final char[] DigitTens;
    static final char[] DigitOnes;
    static final int[] sizeTable;
    public static final int SIZE;
    public static final int BYTES;
    public static java.lang.String toString(int, int);
    public static java.lang.String toUnsignedString(int, int);
    public static java.lang.String toHexString(int);
    public static java.lang.String toOctalString(int);
    public static java.lang.String toBinaryString(int);
    static int formatUnsignedInt(int, int, char[], int, int);
    static int formatUnsignedInt(int, int, char[], int, int);
    public static java.lang.String toString(int);
    public static java.lang.String toUnsignedString(int);
    static void getChars(int, int, char[]);
    static int stringSize(int);
    public static int parseInt(java.lang.String, int) throws java.lang.NumberFormatException;
    public static int parseInt(java.lang.String) throws java.lang.NumberFormatException;
    public static int parseUnsignedInt(java.lang.String, int) throws java.lang.NumberFormatException;
    public static int parseUnsignedInt(java.lang.String) throws java.lang.NumberFormatException;
    public static java.lang.Integer valueOf(java.lang.String, int) throws java.lang.NumberFormatException;
    public static java.lang.Integer valueOf(int);
    public static java.lang.Integer valueOf(long, int) throws java.lang.NumberFormatException;
    public java.lang.Integer(int);
    public java.lang.Integer(long, int) throws java.lang.NumberFormatException;
```

13



```

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.File);
    public java.util.Scanner(java.io.File, java.lang.String) throws java.io.FileNotFoundException;
    public java.util.Scanner(java.io.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.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.Locale 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



```
}
```

### 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

eg:: 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;
```

```
}
```

15



```
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: which ever class has implemented this interface those classes objects are referred as "resources" . 1



18

FileViewImageToolsBrushesShapesColors

ClipboardImageToolsBrushesShapesColors

100%

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
}
// At this line br will be closed automatically
catch(IOException e){
    // handling code
    e.printStackTrace();
}
// handling code
e.printStackTrace();
// handling code
e.printStackTrace();
}
```

Not a good approach

```
FileReader("sample.txt");
```

Problem with writing code for resource releasing

will resolve the problem

try(R)

times

with no change,

code.

20:11:2022

20:11:2022

20:11:2022

20:11:2022

File Edit View Search Document Project Tools Browser Engine Window Help

Directory Content Functions

D:\New Volume

OS

Workspace 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  
8 br = new BufferedReader(new FileReader("output.txt"));  
9  
10 //br.close() will execute automatically and resource will be closed  
11 catch(Exception e){  
12 e.printStackTrace();  
13 }  
14 }  
15 }  
16  
17  
18  
19  
20  
21

18

24°C Clear

Q Search

ln 13 col 10 8 00 PC ANSI

ENG IN

20/11/2022

Nitin Manjrasni

}

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: which ever 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 with in try block.  
try(BufferedReader br=new BufferedReader(new FileWriter("abc.txt")){  
 br=new BufferedReader(new FileWriter("abc.txt"));  
}  
output::CE: can't reassign a value

81



```

}
try(BufferedReader br = new BufferedReader(new FileReader("sample.txt"))){
    br = new BufferedReader(new FileReader("output.txt")); //br.close() will execute automatically and resource will be closed
} catch(Exception e){
    e.printStackTrace();
}

```

**br is final variable**

↓

1. both should be closed
2. output.txt should be closed.
3. confusion whom to close
4. None of the above

such code we call it as "boilerplate" code.

# 20

20-11-2022 classmate - Notes

ExceptionHandling - Nct.

4. untill 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 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.

21



IDE Window

File Edit View Search Run Window Help

Project Explorer

Directory Content Functions

D1 New Volume

Wrappers class

Test.java

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
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  
    }  
}
```

22

Test.java

For Help, press F1

24°C

Cloud

Q Search

Taskbar

17 04:42 29 00 PC ANSI

20:36

30/11/2022

Video Call

Mani Marjan

File Edit View Search Document Project Tools Browser Engine Window Help

Directory Content Functions

D1 New Volume

OS

Wrapper classes

Test.java

4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
public static void main(String[] args) {  
    try(R1)  
    {  
        try(R2){  
            try(R3){  
                }catch (Exception e1){  
            }  
        }catch (Exception e2){  
        }  
    }catch (Exception e3){  
    }  
}
```

Ln 17 Col 78 27 29 PC ANSI ENG IN 20:38 26-11-2022

Sanjay

23

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){  
            }  
        }  
    }  
}
```

2x



Editor - (D:\workspace\test.java)

File Edit View Search Document Project Tools Browser Engine Window Help

Directory Content Functions

D:\New Volume

Workspace

test.java

7  
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 occurred and resulted in exception and if we don't have
handling logic in our application and if the exception object delegated to
JVM(default exception handler) then it is abnormal termination
*/
}
}
```

1234567

Sanjay

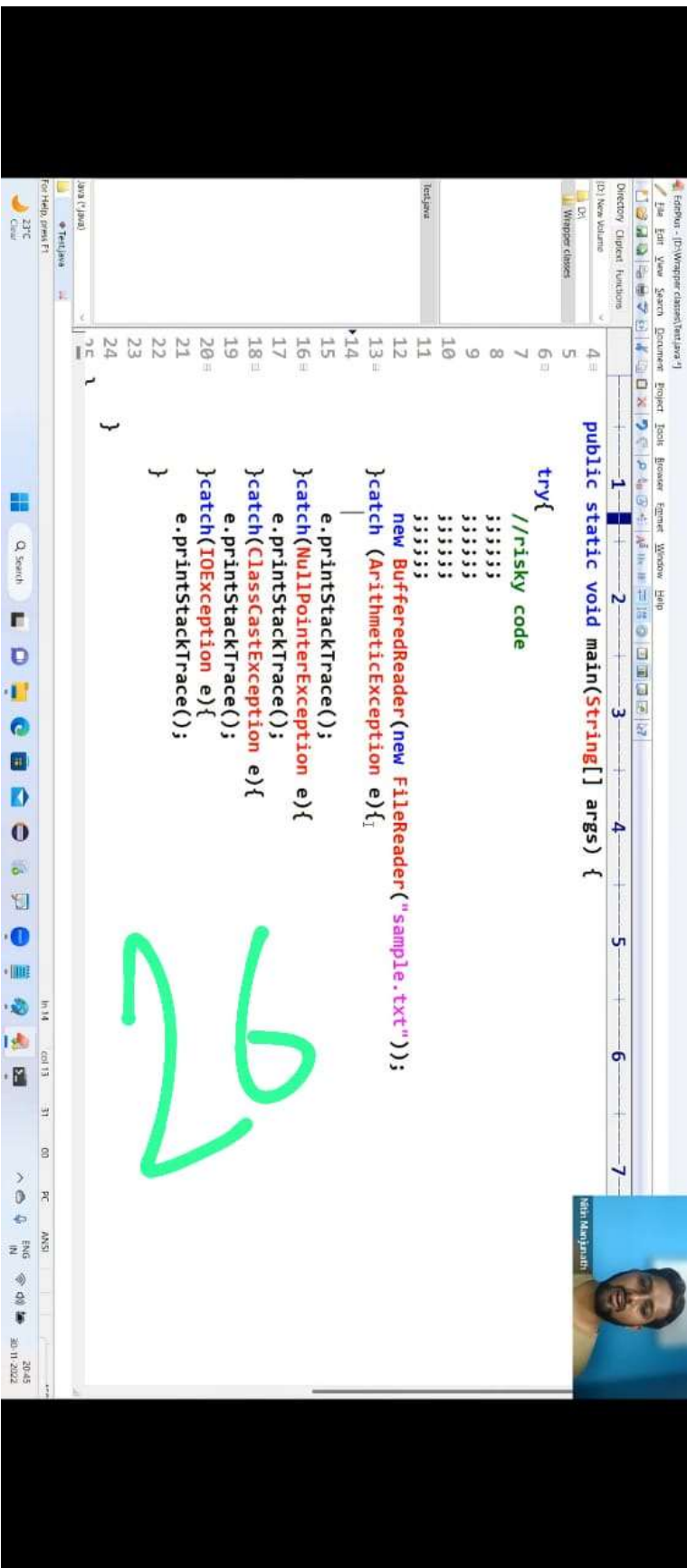
24°C Clear

Q Search

19 18 00 75 27 00 PC ANSI

20:42 30-11-2022

25



Editor - (D:\workspace\test.java)

File Edit View Search Runway Project Tools Browser Engine Window Help

Directory: C:\Program Files\Java\jdk-11.0.10\bin

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 (ArithmeticException | NullPointerException e){

14 //handling code

15 e.printStackTrace();

16 }catch (ClassCastException | IOException e){

17 //handling code

18 e.printStackTrace();

19 }

20 }

21 }

22 }

Ln 5 Col 46 28 00 PC ANSI

20:48

30/11/2022

23°C

Cloud

Q Search

Taskbar icons: File Explorer, Edge, Task View, etc.

Video call window: Manoj Kumar

Handwritten green mark: A large green '5' or '7' shape.

## MultCatchBlock

=====

Till jdk1.6, eventhough 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());
}
```

28





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

20



Ln: 726 Col: 1  
23°C  
Clear

Q Search



Windows [CTRL]  
ENG IN UTT 5  
20:50  
20-11-2022

```

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
15        }catch (ArithmeticException | Exception e){
16            //handling code
17            e.printStackTrace();
18        }
19    }
20 }
21 }
22

```

3



Mr. Manjiv



Mr. Manjiv



Mr. Manjiv



Mr. Manjiv



Mr. Manjiv

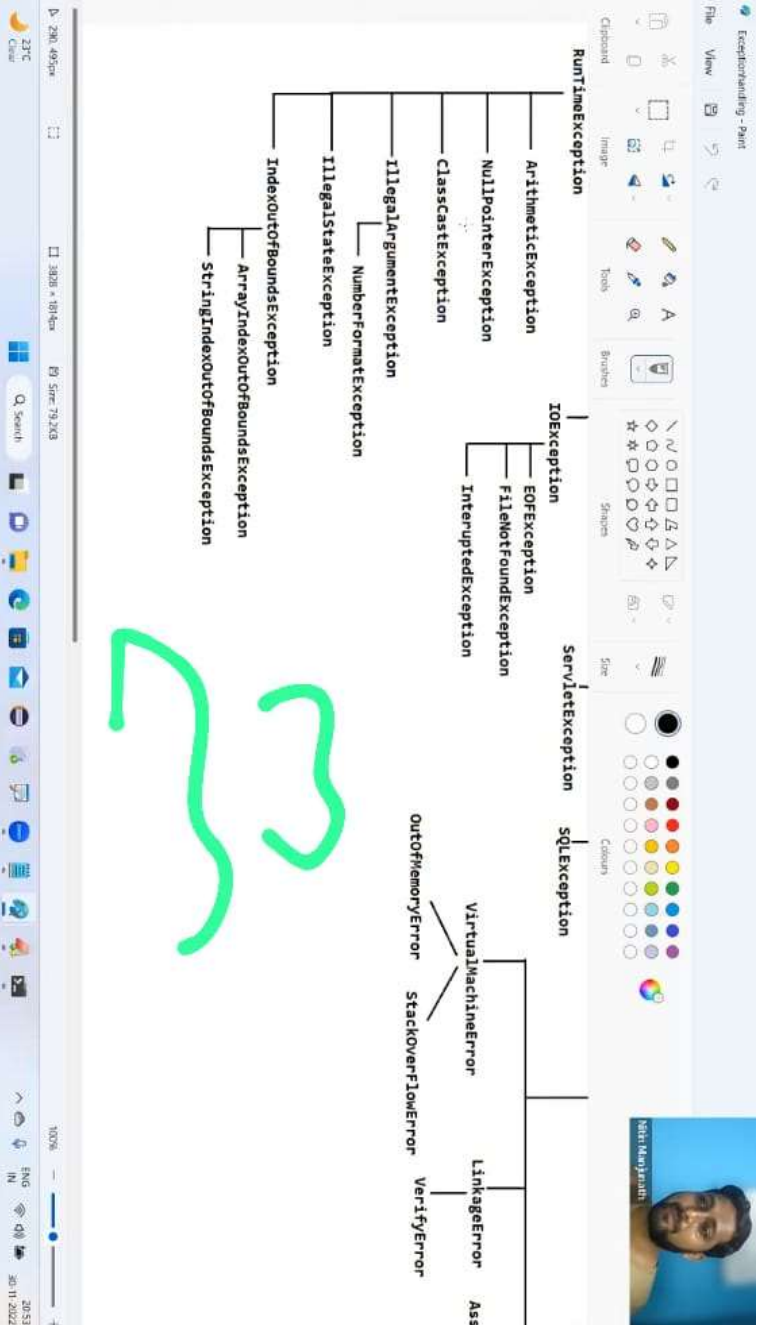


Mr. Manjiv



Mr. Manjiv

52



IntelliJ IDEA - WiproExamTest.java

File Edit View Search Database Project Tools Browser Editor Window Help

Directory: C:\Users\... \src\main\java\WiproExamTest

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 (ArithmeticException | NullPointerException | IOException | ClassNotFoundException  
18 //handling code  
19 e.printStackTrace();  
20 }  
21 }  
22 }

Ln 15 Col 34 29 29 PC ANSI 20:56 30-11-2022

34

WiproExamTest

Command Prompt

X

+

-

D:\Wrapper\classes>javac Test.java

D:\Wrapper\classes>javac Test.java

Test.java:17: error: exception ClassNotFoundFoundException is never thrown in body of corresponding try statement  
 }catch (ArithmeticException | NullPointerException | IOException | ClassNotFoundFoundException e){  
 ^


1 error

D:\Wrapper\classes>javac Test.java

D:\Wrapper\classes>


25

Nitin Manjappa



23°C  
Clear

Q Search



ENG  
IN

20:56  
20-11-2022

## Exception Handling

- =====
- 1. try with resource.
- 2. try with multi-catch block.
- 3. Rules of Overriding associated with Exception.

### Remaining topics to be discussed

- =====
- 1. instanceof vs instanceof(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 multcatch block

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

36



### Rules of using try with resource

=====

1. We can declare any no of resources, but all these resources should be separated 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

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

```
public interface java.lang.AutoCloseable {
```

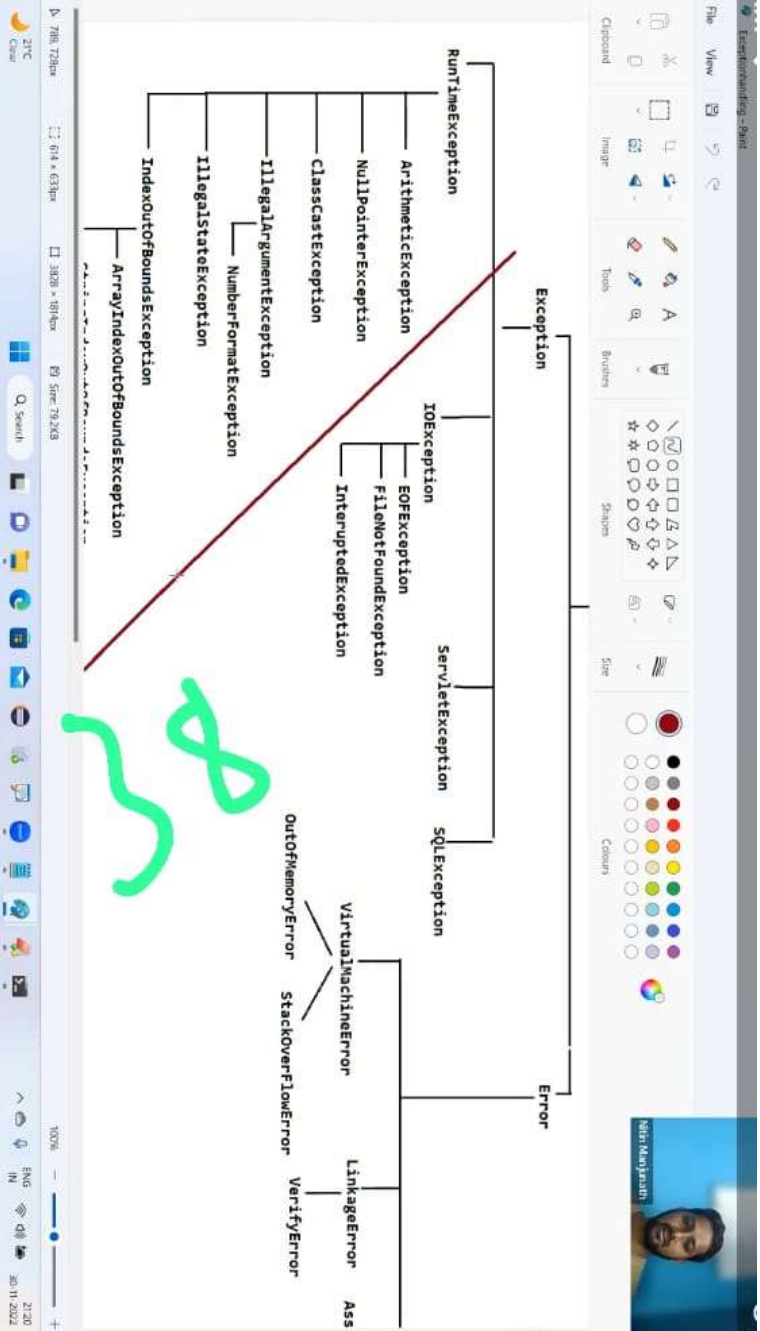
```
public abstract void close() throws java.lang.Exception;
```

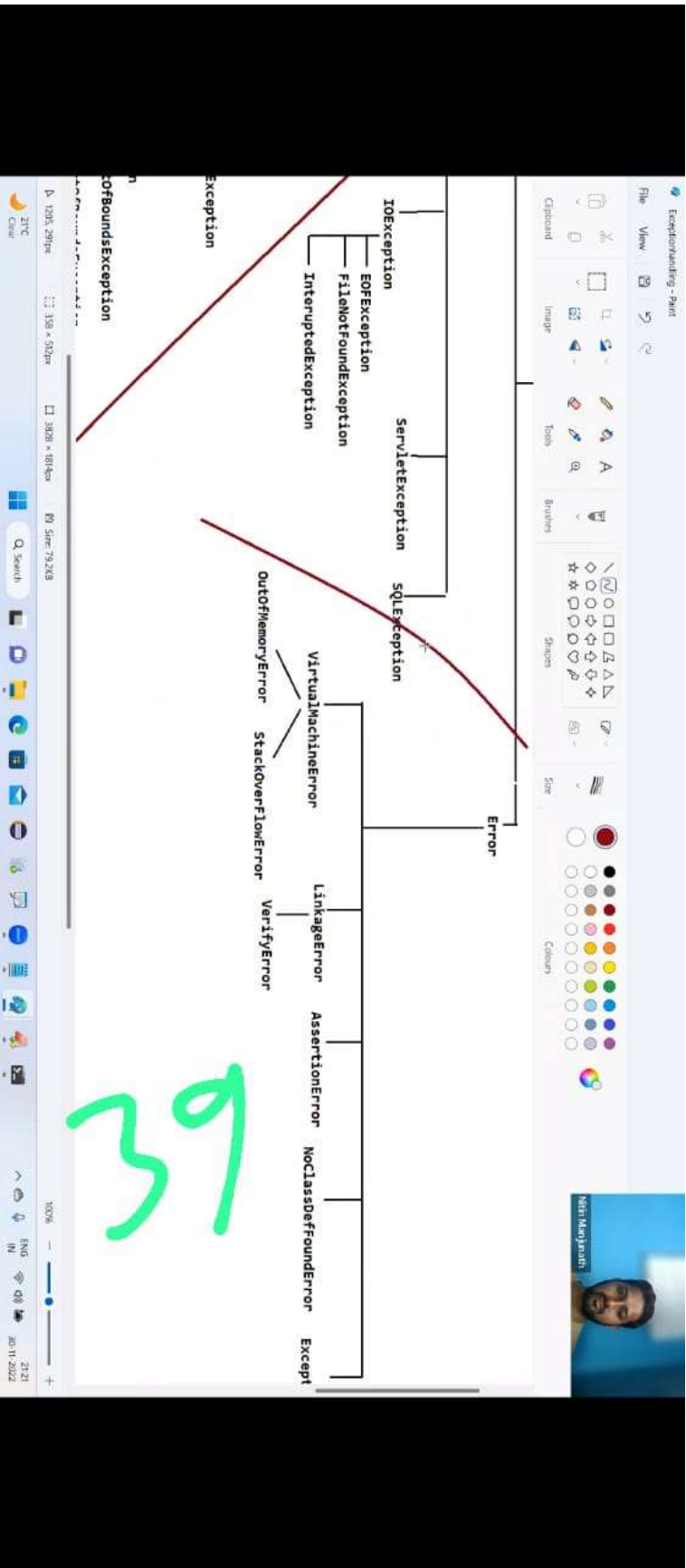
```
}
```

Note: which ever 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 with in try block.

37





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

40



File Edit View Search Run Window Help

Project Tools Browser Engine Window Help

Directory Content Functions

D:\New Volume

Wapper class

Test.java

1 import java.io.\*;  
2  
3 class Parent  
4 {  
5 public void m1() throws Exception {  
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 }

Ln 13 Col 34 28 70 PC ANSI

27:26 26/11/2022

Mini Map

7

### 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 the same exception.

There are no restrictions on UncheckedException.

eg#1.

```
class Parent{  
    public void methodOne();  
}  
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

72



### 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





```
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,ArithmeticException{}  
output: invalid
```

45



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

```
parent: public void methodOne()throws IOException{
```

```
child : public void methodOne()throws FileNotFoundException,ArithmeticException{  
output: valid
```

```
parent: public void methodOne()
```

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

46



output: valid

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

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

47



instanceof

=====

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

r instanceof X

r => reference

X => class/InterfaceName

eg:

ArrayList al = new ArrayList();

al.add(new Student());

al.add(new Cricketer());

al.add(new Customer());

48



30-11-2022: classmate - Notepad

File Edit View

```
al.add(new Student()); // 0th position  
al.add(new Cricketer()); // 1st position  
al.add(new Customer()); // 2nd position
```

Object o=al.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

}

5



Ln 237, Col 34

27°C  
Clear

Q Search



100%

Windows [CTRL]

UT 8

ENG IN

27:45  
30-11-2022

30-11-2022 7:47 AM resource - Start

FileView

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

class Thread extends Object implements Runnable(  
@Override  
public void run()  
{  
}  
}

Object(C)  
Runnable(I)  
Thread(C)

3. confusion whom to close  
4. None of the above

50

100%

ENG

21:47

30-11-2022

Sanjay

Editor - (D:\workspace\test.java)

File Edit View Search Runway Project Tools Browser Tagman Window Help

Directory Content Functions

D:\workspace

test.java

1 import java.io.\*;

2

3

4

5 public class Test {

6 public static void main(String[] args) {

7 Thread t = new Thread();

8 System.out.println(t instanceof Object); //true

9 System.out.println(t instanceof Thread); //true

10 System.out.println(t instanceof Runnable); //true

11 System.out.println(t instanceof String); //false

12 }

13 }

14

15

16

17

18

19

Test.java


For Help, press F1

Q Search

Ln 11 Col 31 19 00 PC ANSI

27:50 26/11/2022

Nitin Mahajan



Editor - (D:\workspace\test.java)

File Edit View Search Document Project Tools Browser Engine Window Help

Directory Content Functions

D:\New Volume

OS

Workspace

Workspace classes

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

```
1 import java.io.*;
2
3
4
5 public class Test {
6     public static void main(String[] args) {
7         Object t = new Thread();
8         System.out.println(t instanceof Object);//true
9         System.out.println(t instanceof Thread);//true
10        System.out.println(t instanceof Runnable);//true
11        System.out.println(t instanceof String);//false
12        System.out.println(t instanceof null);//false
13    }
14 }
15
16
17
18
19
20
```

Java 17 (JDK)

Test.java


For Help, press F1

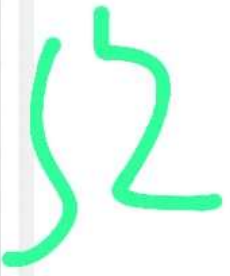
27°C Kolkata, W

Q Search

Ln 12 Col 45 20 29 PC ANSI ENG IN 20/11/2022

Nitin Dasgupta





```
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



=> 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  
    }  
}
```

JS



=====

### Difference between instanceof and isInstance() :

instanceof

=====

instanceof 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()

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.



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

isInstance()

isInstance() is a method, present in class Class, we can use isInstance() method to check 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
```

SS





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
}
```

75

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



Q>

Given this code from Class B:

25. A a1 = new A();

26. A a2 = new A();

27. A a3 = new A();

28. System.out.println(A.getInstanceCount());

What is the result?

1. public class A{

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. }

BS



```
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

1

60



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



File

View

Image

Tools

Brushes

Shapes

Size

Colours

Clipboard

Image

Tools

Brushes

Shapes

Size

Colours

30-11-2022 19:46:19 resource - Paint

20°C

400 x 197px

3490 x 3881px

File

Search

Taskbar

100%

22:54

30-11-2022

```
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);

// insert code here 29
System.out.println(o.getX());
```

```
class Outer {
    private Inner y;
    public void setY( Inner y ){ this.y = y; }
    public Inner getY(){ return y; }
}
```

Outer

x = n

x = 10

Inner

n = 10

29

B. i.setX( 100 );

C. o.getX().setX( 100 );

D. i = new Inner(); i.setX( 100 );

E. o.sety( i ); i = new Inner();

F. i = new Inner(); i.setX( 100 );

G. i = new Inner(); i.setX( 100 );

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

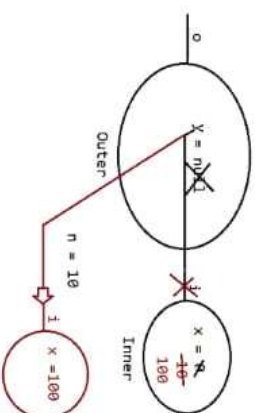


```
Outer o = new Outer();
Inner i = new Inner();
int n = 10;
i.setX(n);
o.setY(i);
```

answer: produce the output 100?

```
// 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 ); ✓



63

50-11-2022\_201part\_session - Notepad

File Edit View

public class Gamma {  
 public static void main(String[] args) {  
 Outer o = new Outer();  
 Inner i = new Inner();  
 int n = 10;  
 i.setX(n);  
 o.setX(i);  
 // insert code here 29  
 System.out.println(o.getX().getX());  
 }  
}

A. n = 100;  
B. i.setX( 100 );  
C. o.getX().setX( 100 );  
D. i = new Inner(); i.setX( 100 );  
E. o.setX( i ); i = new Inner(); i.setX( 100 );  
F. i = new Inner(); i.setX( 100 ); o.setX( i );

Answer: BCF

64

Nitin Kharjane

Ln 102, Col 1

20°C Clear

Q Search

100%

Windows [CTRL]

ENG IN

UTZ 5

22:40 30-11-2022

FileEditView

\*50-11-2022\_oopppt\_session - Notepad

```
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. foofoofoofoo


C. foofoofoofoo

D. foofoofoofoo

E. barbarbarbar

65

Nar Karjane



100%

Windows [CTRL]

22:43

30/11/2022

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. foofoofoofoofoo
- B. foobarfoobarbar
- C. foobarfoofoofoo
- D. foobarfoofoofoo
- E. barbarbarbarbar
- F. foofoofooobarbar
- G. foofoofooobarfoo

Answer: D

1

Ln 178, Col 1  
20°C  
Clear

Q Search



100%

Windows (CTRL)

| UTT 5

ENG  
IN 22:44  
30.11.2022



The screenshot shows a Windows desktop environment. In the background, there are several application icons on the taskbar, including File Explorer, Edge browser, and various utility tools. The foreground features the Visual Studio Code editor window.

**Visual Studio Code Editor:**

- File Explorer:** Displays the contents of the "src" folder, which includes files like "Hello.java", "Main.java", and "Test.java".
- Code Editor:** Contains the following Java code snippet:
 

```
public class Hello {
    String title;
    int value;
    public Hello() {
        title += " World";
    }
    public Hello(int value) {
        this.value = value;
        title = "Hello";
    }
    Hello();
```
- Output Console:** Shows the execution output:
 

```
Hello c = new Hello(5);
System.out.println(c.title);
```
- Run and Debug View:** Displays the "Hello" method call stack, indicating it results in "CE".
- Large Green Number:** A large, hand-drawn green number "68" is overlaid on the bottom right corner of the screen.

30-11-2022\_snippets\_session - Notepad

File Edit View

```
this.value = value;
this.value = value;
```

```
title = "Hello";
```

```
Hello();
```

}

—

```
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

LITTON Coal

 20°C  
Clear Q Search

D



1042



2

10096

Windows (ICM)

ENGL  
IN

10

1

22

1

Nathan Aspinwall