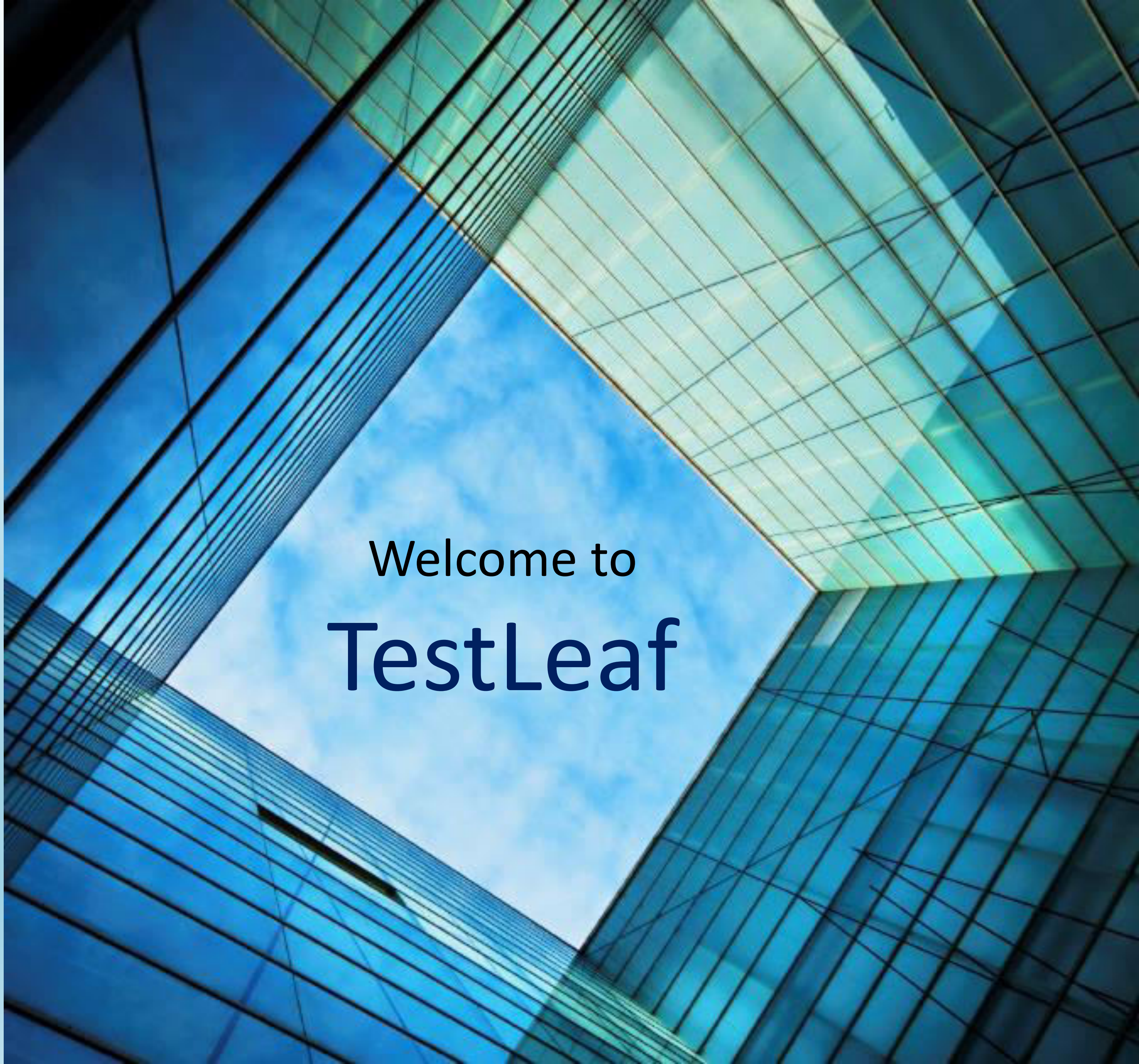
Exception Handling & Eclipse Debugging

TestLeaf



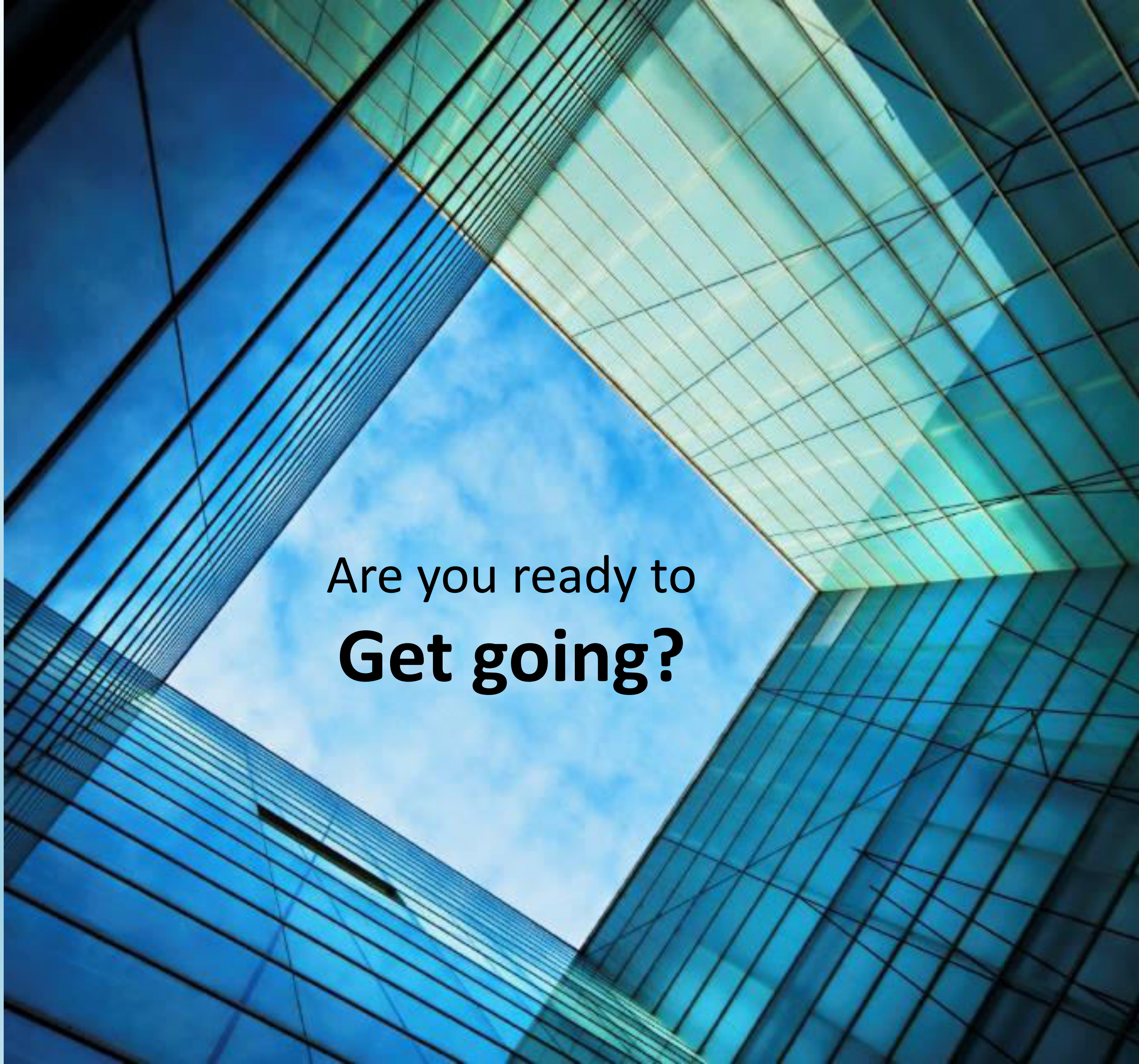
Welcome to

TestLeaf

Agenda - What We'll Cover Today

- What is Exception in Java / Selenium and how to handle it
- How to debug your code in eclipse with all short keys

Exception Handling



Are you ready to

Get going?

Definition of exception and exception handling

Exception:

Exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. It *doesn't stops your program* from running.

1. Checked Exception (Compile time)
2. Unchecked Exception (Run time)

Exception Handling:

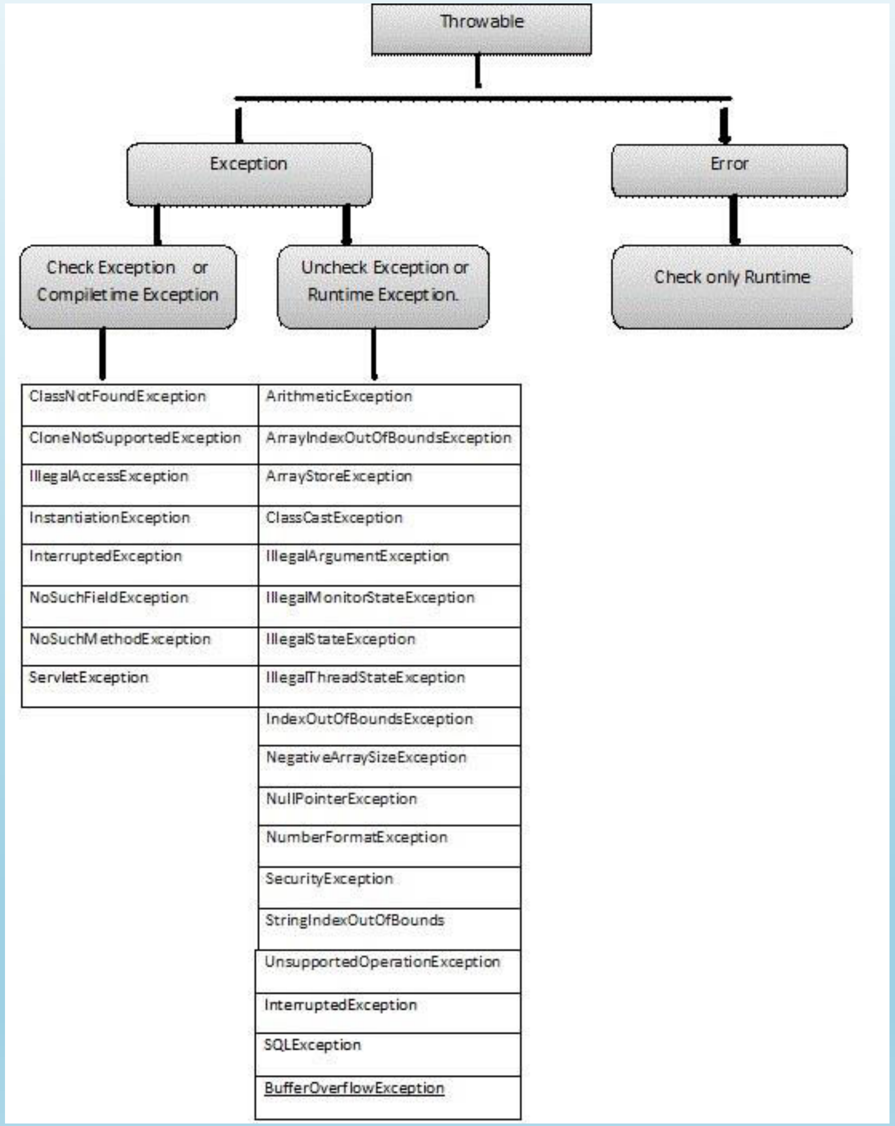
Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException,IOException, ArrayIndexOutOfBoundException etc.

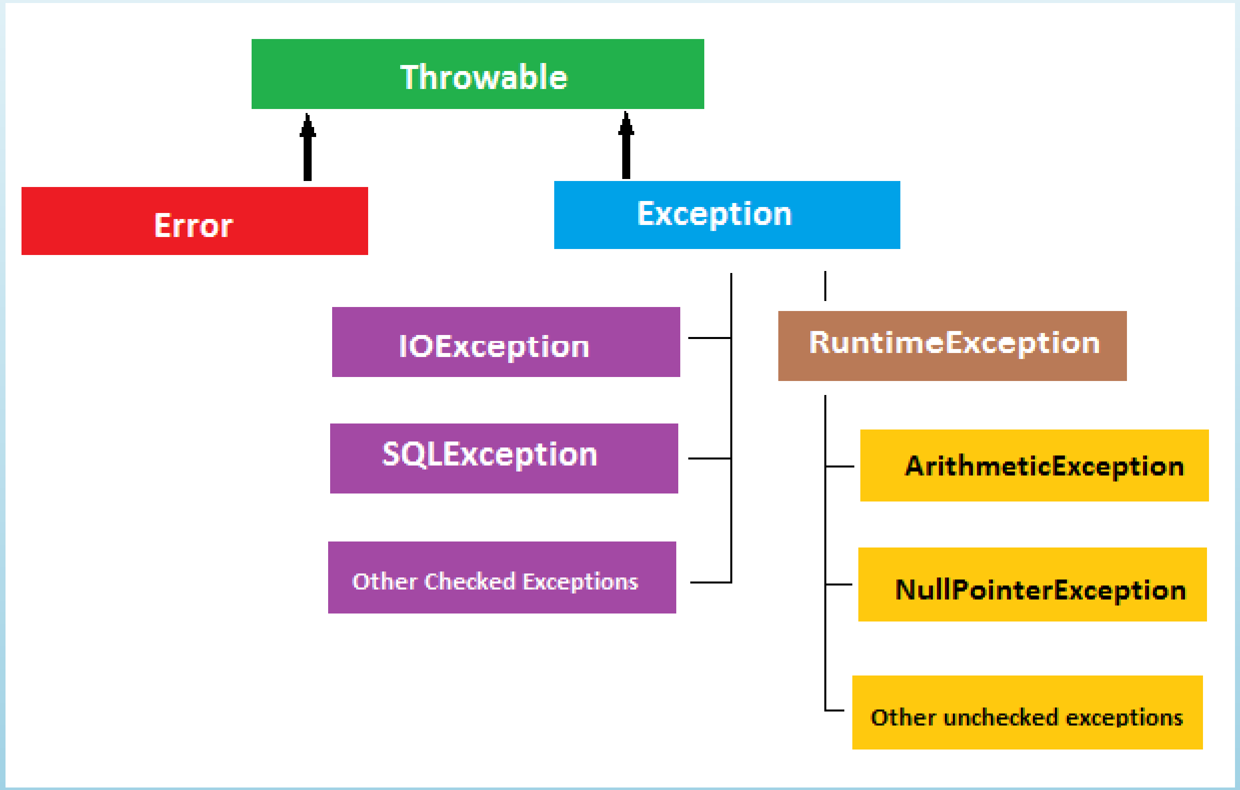
Error:

Error is extend from Throwable class and it *stops the program to continue* and we can't handle it.

E.g. outofMemoryError, StackOverFlowError

Java Exception





Try , Catch Finally

- Try – is the block where your real logic or implementation exists.
- Catch – is the block will catch when exception occurs at run/compile time.
- Finally – is the block irrespective of exception will execute always

How finally works in exception handling

- Good example to recollect Finally

➢ Will get mark sheet even we didn’t get pass mark in subject.

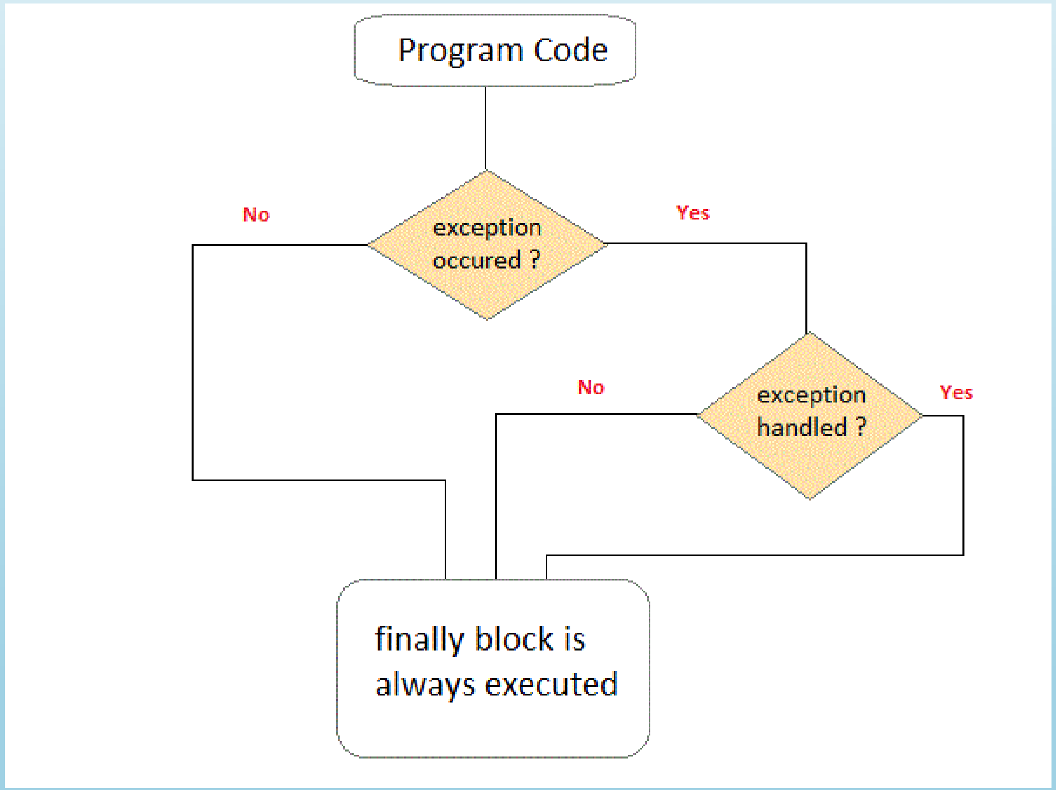
➢ Time always run even we don’t utilized.

- In selenium where can we use finally block

➢ Take screen shot after each action.

➢ Close or remove unused reference object after the action like closing FileInputStream, XSSFWorkbook etc.

How finally works in exception handling



Syntax of try catch and finally

```
try{

}catch(Team – Lead - Exception e1){

}catch(Project – Manager - Exception e2){


}

.....
.....

catch(CEO - Exception en){

}finally{

}
```

Exception class should catch from low to high like in this example

How exception handling in selenium

- *In selenium all the exception are “**RUN TIME EXCEPTON**”*
- Whenever you are handling with driver object, web element it always recommended to handle the exception with try , catch and finally.
- Few e.g. of selenium exception

    - No such element found exception

- java.lang.Object
  - java.lang.Throwable
    - java.lang.Exception
      - java.lang.RuntimeException
        - org.openqa.selenium.WebDriverException
          - org.openqa.selenium.NotFoundException
            - org.openqa.selenium.NoSuchElementException

Selenium Exception

What is throws in Exception Handling

- Throws is a keyword will be declared in method level
- When there is any compile time exception occurs and you don’t want to handle in the same class / method use throws
- Throws will throw the exception to calling method, where you are forced to handle with either try catch or throws
- If we throws exception no need to write try, catch and finally blocks
- E.g.. FileNotFound Exception, Thread.sleep()
- Syntax:
- *public void getName(String employeeName) throws FileNotFoundException, Exception {*

```
        // logic
        }
```

Note: You can throws more than one Exception class, but all should be catch from calling method.

Difference between throw and throws in Exception handling

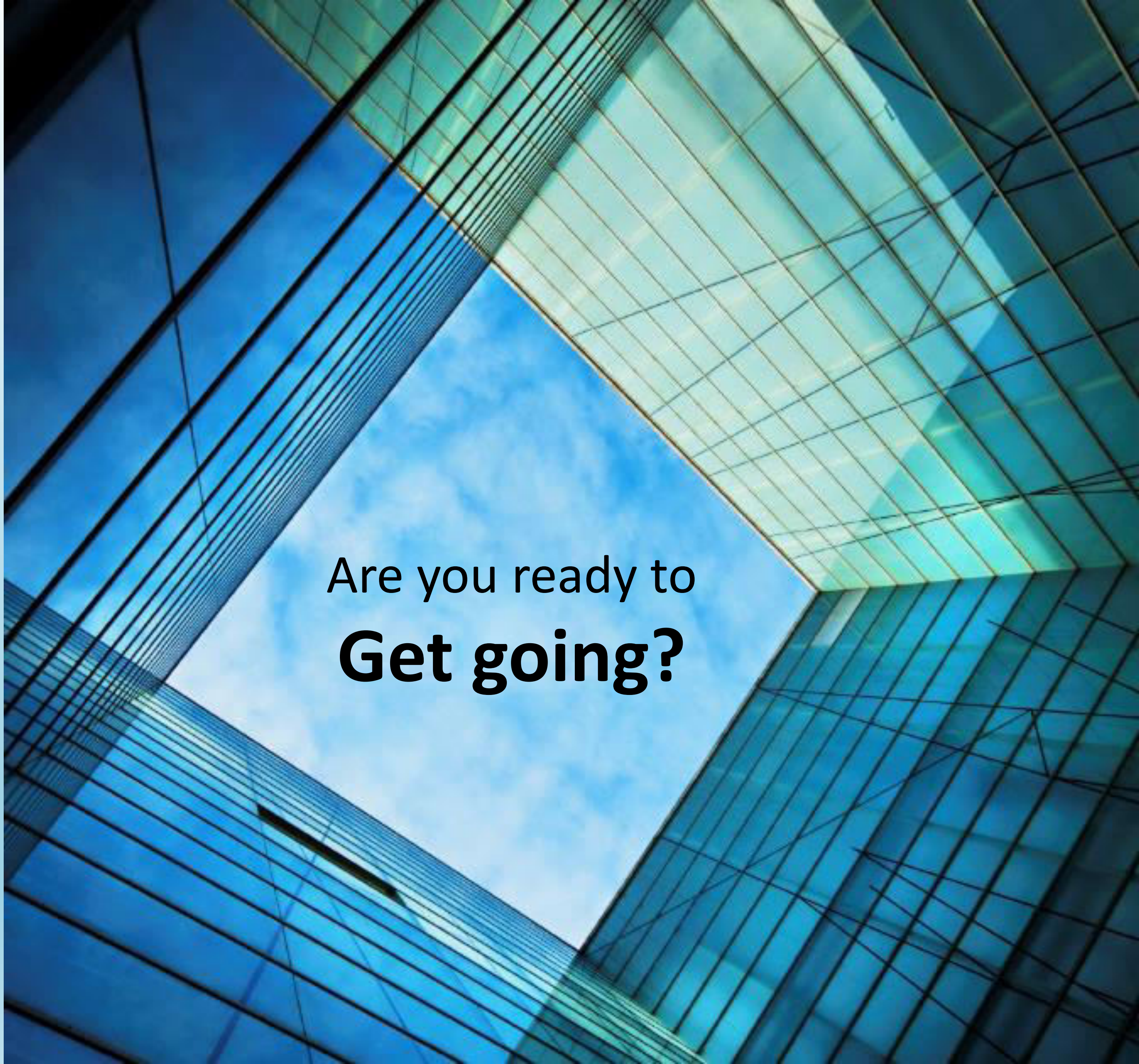
Throws	Throw
Used in method level signature	Used inside try or catch block, even without try catch
Throws to calling method	Throw to specified exception class
Your program continues run even there is exception occurs	Your program stops and exit from the program

Syntax: method throws Exceptoin {
    // method implementation
}
	Syntax: throw **new** Exception class Constructor();

Questions

1. What are keywords are used in exception handling ?
2. Which block is mandatory in exception handling try, catch or finally
3. Method should throws only one exception class ?
4. How do you create user defined exception class ?
5. Eclipse by default will give suggestion to handle checked exception or unchecked exception or both ?
6. Is this good practice to handle run time exception by **throws** ?
7. Who is super most class of exception handlings ?
8. How many types of exception do we have ?
9. Which one stops to continue to execute program either Exception or Error ?
10. Does Error can be handled by programmatically ?

Debugging in Eclipse



Are you ready to

Get going?

Debugging is a in build feature coming with eclipse and net beans, to debug your code at run time..

<u>Features :</u>

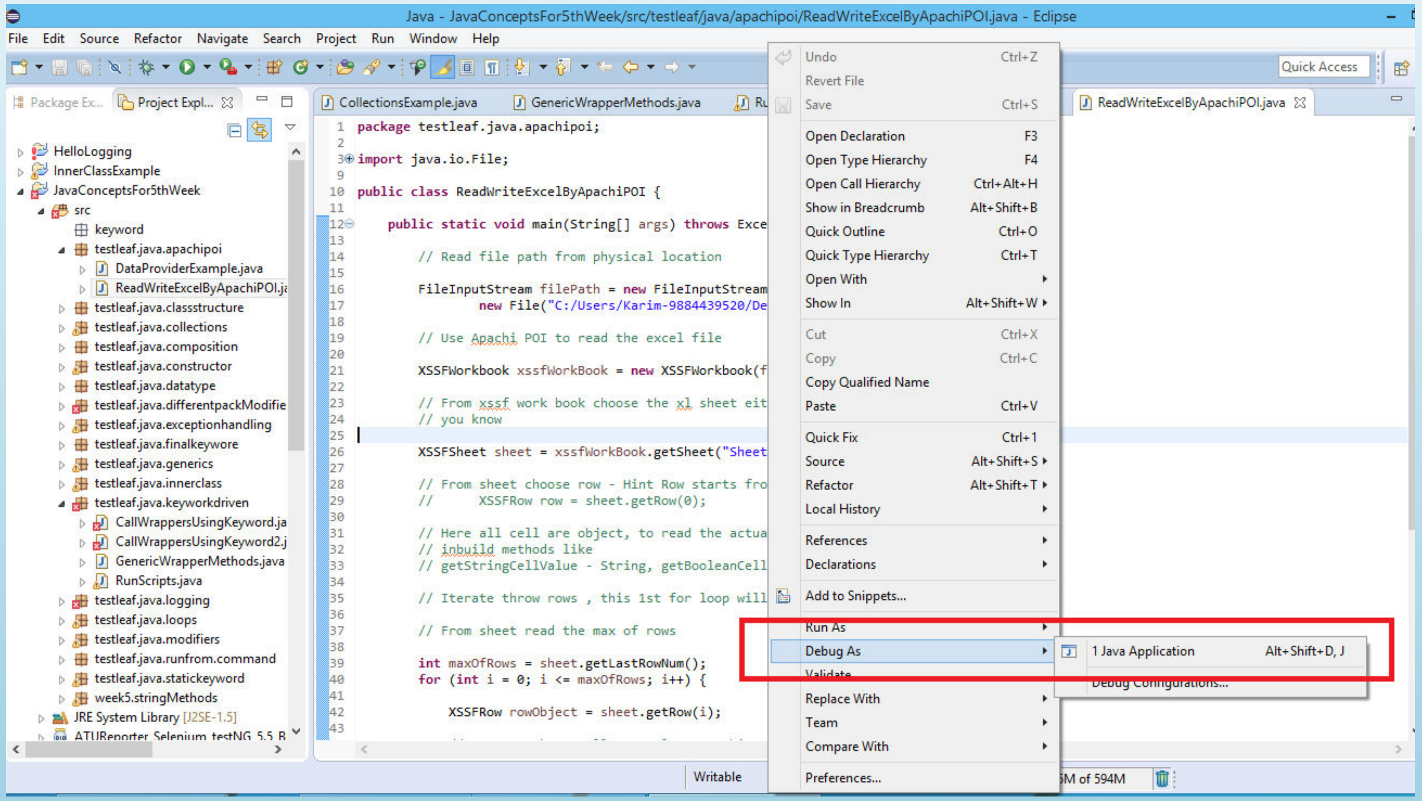
- Pass your code at break point
- Can read / change the values of object at break point
- Can watch the variable

Key	Functionality
F5	Step into
F6	Next line of code
F8	Next Break point

Short cuts keys

How to run the code in Debug mode

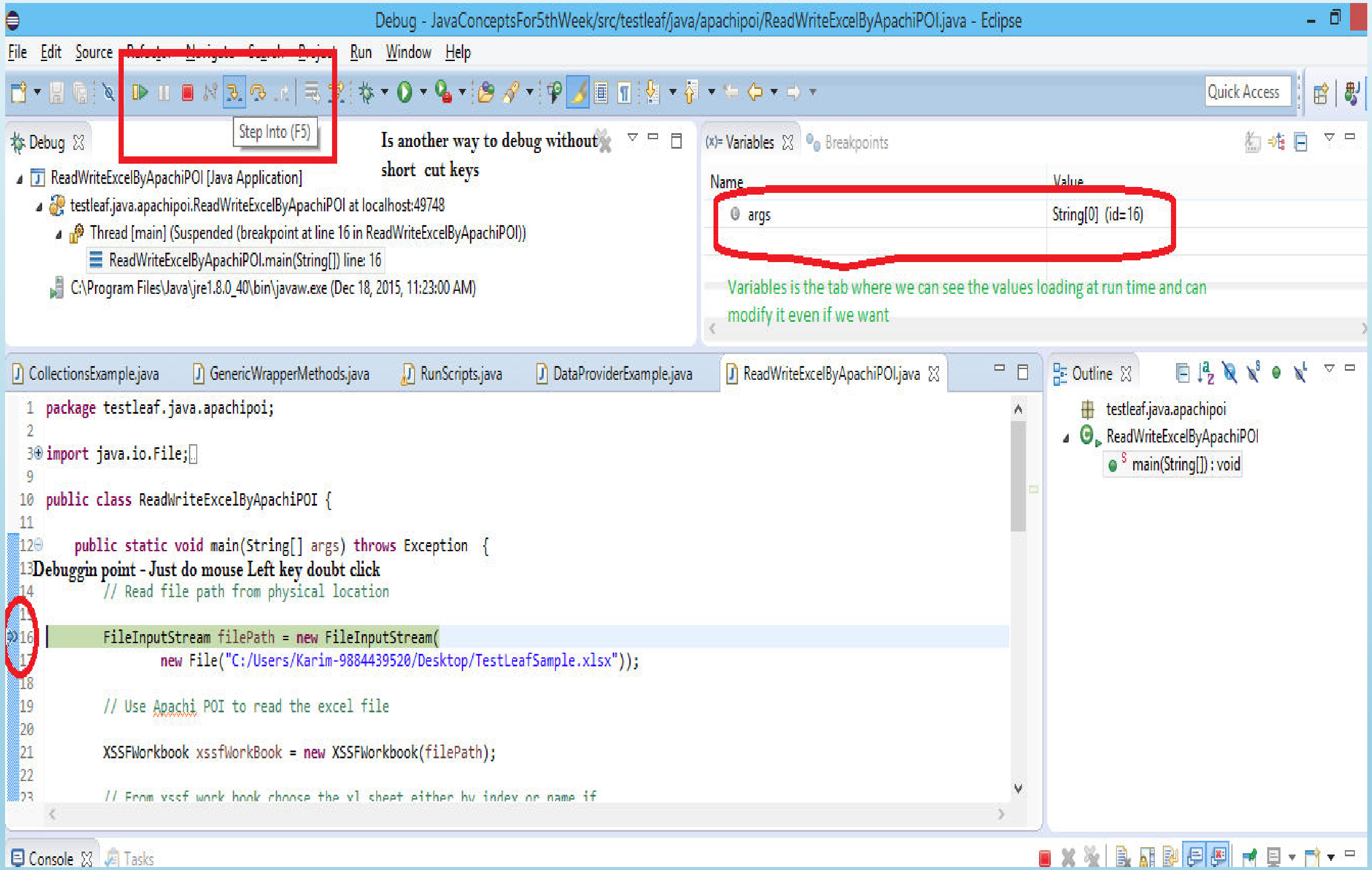
Right click on java file, choose Debug As option rather Run As option


Java - JavaConceptsFor5thWeek/src/testleaf/java/apachipoi/ReadWriteExcelByApachiPOI.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Ex... Project Expl...
HelloLogging
InnerClassExample
JavaConceptsFor5thWeek
src
keyword
testleaf.java.apachipoi
DataProviderExample.java
ReadWriteExcelByApachiPOI.ja
testleaf.java.classstructure
testleaf.java.collections
testleaf.java.composition
testleaf.java.constructor
testleaf.java.datatype
testleaf.java.differentpackModifie
testleaf.java.exceptionhandling
testleaf.java.finalkeywore
testleaf.java.generics
testleaf.java.innerclass
testleaf.java.keyworddriven
CallWrappersUsingKeyword.ja
CallWrappersUsingKeyword2.j
GenericWrapperMethods.java
RunScripts.java
testleaf.java.logging
testleaf.java.loops
testleaf.java.modifiers
testleaf.java.runfrom.command
testleaf.java.stackeyword
week5.stringMethods
JRE System Library [J2SE-1.5]
ATUReporter Selenium testNG 5.5 B
CollectionsExample.java
GenericWrapperMethods.java
ReadWriteExcelByApachiPOI.java
Undo Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Add to Snippets...
Run As
Debug As
Validate
Replace With
Team
Compare With
Preferences...
1 Java Application Alt+Shift+D, J
Debug Configurations...
Writable
M of 594M

```
package testleaf.java.apachipoi;

import java.io.File;

public class ReadWriteExcelByApachiPOI {

    public static void main(String[] args) throws Exce

        // Read file path from physical location

        FileInputStream filePath = new FileInputStream
                new File("C:/Users/Karim-9884439520/De

        // Use Apachi POI to read the excel file

        XSSFWorkbook xssfWorkBook = new XSSFWorkbook(f

        // From xssf work book choose the xl sheet eit
        // you know

        XSSFSheet sheet = xssfWorkBook.getSheet("Sheet

        // From sheet choose row - Hint Row starts fro
        //      XSSFRow row = sheet.getRow(0);

        // Here all cell are object, to read the actua
        // inbuild methods like
        // getStringCellValue - String, getBooleanCell

        // Iterate throw rows , this 1st for loop will

        // From sheet read the max of rows

        int maxOfRows = sheet.getLastRowNum();
        for (int i = 0; i <= maxOfRows; i++) {

            XSSFRow rowObject = sheet.getRow(i);
```

Other tabs in Debug window

- Without short cuts also we can debug our code
- Run time we can read / modify the values of variable in Variable tab



Debug - JavaConceptsFor5thWeek/src/testleaf/java/apachipoi/ReadWriteExcelByApachiPOI.java - Eclipse

Is another way to debug without short cut keys

```
ReadWriteExcelByApachiPOI [Java Application]
  testleaf.java.apachipoi.ReadWriteExcelByApachiPOI at localhost:49748
    Thread [main] (Suspended (breakpoint at line 16 in ReadWriteExcelByApachiPOI))
      ReadWriteExcelByApachiPOI.main(String[]) line: 16
  C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (Dec 18, 2015, 11:23:00 AM)
```

Name	Value
args	String[0] (id=16)

Variables is the tab where we can see the values loading at run time and can modify it even if we want

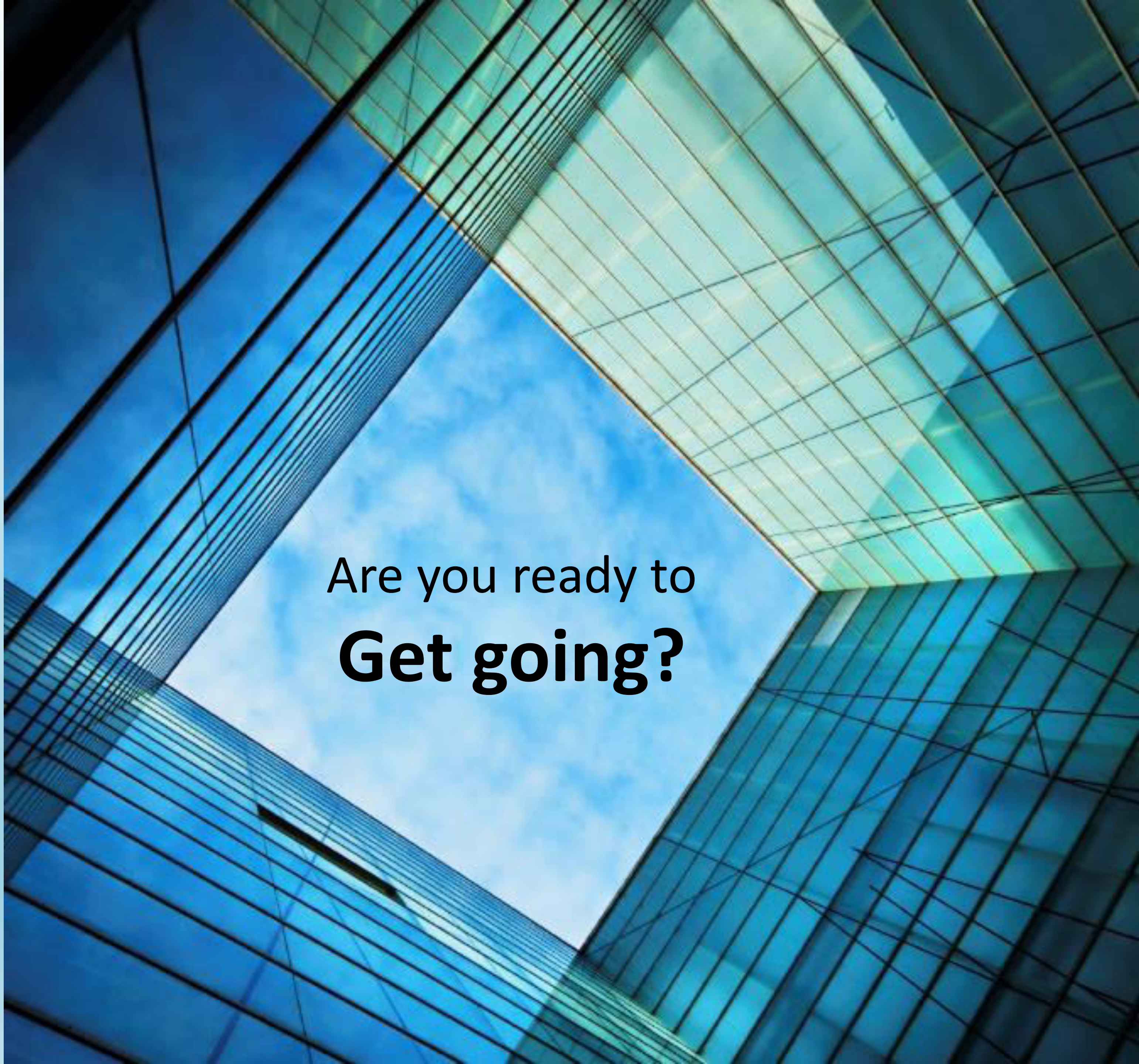
CollectionsExample.java | GenericWrapperMethods.java | RunScripts.java | DataProviderExample.java | ReadWriteExcelByApachiPOI.java

```
package testleaf.java.apachipoi;

import java.io.File;

public class ReadWriteExcelByApachiPOI {

    public static void main(String[] args) throws Exception  {

        // Read file path from physical location

        FileInputStream filePath = new FileInputStream(
                new File("C:/Users/Karim-9884439520/Desktop/TestLeafSample.xlsx"));

        // Use Apachi POI to read the excel file

        XSSFWorkbook xssfWorkBook = new XSSFWorkbook(filePath);

        // From xssf work book choose the xl sheet either by index or name if
```

Debuggin point - Just do mouse Left key doubt click

```
testleaf.java.apachipoi
  ReadWriteExcelByApachiPOI
    main(String[]) : void
```

Recap



Are you ready to

Get going?

- Exception Handling

  - ✓ Try catch finally are the key words will be used in exception handling.
  - ✓ Selenium exception all are run time exception.
  - ✓ Run time exception all should surround with try catch rather throws.
  - ✓ All exception are extends to Exception class by default.
  - ✓ We can have nested try catch block and more than one catch block
  - ✓ Throws exception can be more than one

- Debugging in eclipse

  - ✓ F5 , F6, F8 are short cuts are used for debugging.
  - ✓ On debugging we can read / modify the variables at run time.

Summary

Q&A

- Seleniumhq.org
- JavaTpoint
- Wikipedia
- http://selenium.googlecode.com/git/docs/api/java/index.html

Reference Links and Books

Thank you

Hope you had a
Great Learning!