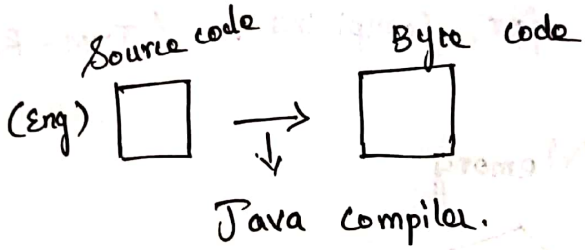
13/04/2019 Java

Java :-

* Simple programming language.
* Writing, compling & debugging is easy.
* It helps to reuse.

FEATURES OF Java :-

1. Platform Independent.
2. Open Source
3. Multi - threading
4. Secure
5. Portable



Java compiler converts the Source code into Byte code.

Source code :- Java code

Byte code : (.class file).

1. platform Independent: Any platform Supports like windows, linux, mac, etc.
2. open Source : Free of cost.
3. Multi Threading : We can execute more than one task parallely or Simultaneously.
4. Secure : There is a "Virtual firewall" between the computer & application. So it will not allow any unauthorized access.
5. portable : Write once, run anywhere.

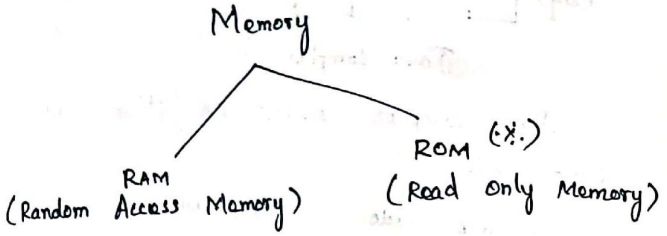
Java Architecture :-

JDK - Java Development Kit.
JRE - Java Runtime Environment.
JVM - Java Virtual Machine.

JDK :- We have to download java development kit or tool to run the java code.
Version => 1.6 to 1.11 (current usage : 1.8 Version)

JRE :- It has (.class file). It is a predefined files.

JVM :- It allocates memory for all tasks.
It helps for compiling. (JVM - RAM)



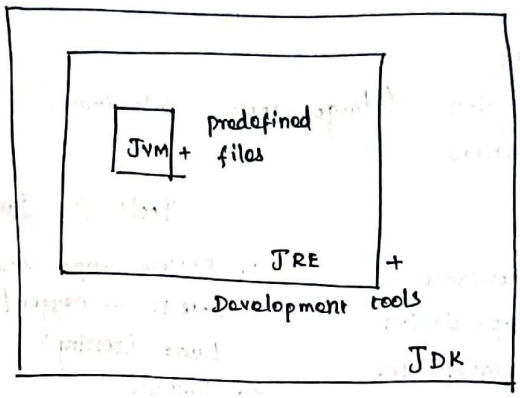
1. Volatile
2. We can delete the memory.

1. We cannot delete the memory.

Programs are storing in "Heap Memory" in RAM.

Garbage Collection :-

Unwanted or unused memory will be destroyed automatically. It is not possible in c, c++.



CODING STANDARD :-

1. Pascal Notation
2. Camel Notation

Eg : employee details

1. Employee Details
First letter will be capitalized.
2. No space b/w 2 words
3. We can seperate the words using underscore (_)
4. Project Name & Class Name

1. employee Details
First letter of 1st word will be in small letter & the 1st letter of the following word will be in capital letter.
2. Object Name
Method Name
Variable Name.

Java Concepts :-

1. OOps concepts
2. Constructions or
3. Control Staroment
4. Strings
5. Arrays
6. Collections (Large topic - Selenium).
7. Exceptions

① Oops :-

a. Inheritance
b. Encapsulation
c. Polymorphism
d. Abstraction

Tools for Java

1. Eclipse - open soure - 2018 12 --- oxygen / phoon - Luna (testing)
2. Intellji
3. Net beans
4. Notepad
5. IBM RAD.

oops :- Object Oriented Programming Structure

, The way of Implementation in which programs are organized in the form of objects, methods & Class.

1. Object : Instance of the class.
   It allocates Memory.
   Used to call methods.

Synrax :

Class Name Object = New ClassName();
(Variable Name)

Example :

```
Class Student {
    ....
    .....
}

Student St = New Student ();
obj. Name
```

2. Methods :- Set of actions to be performed.

```
Class Student {
    .....
    .....
    Student Name() {
        .....
        .....
    }
}
```

=> methods

Syntax :-

St. Student Name () => Calling methods using objects.

3. Class :-

Combination of objects and methods.

OOps concepts :-

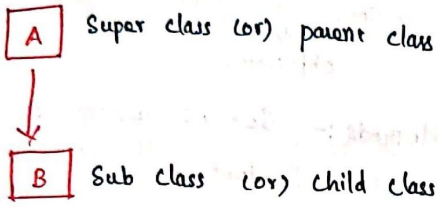
① Inheritance :

a) Single
b) Multiple
c) Multi - level
d) Hierarchical
e) Hybrid

a) Single Inheritance :-

One parent class is accessed by one child class.

Example:



Accessing property of one class into another class using "Extends" Keyword.

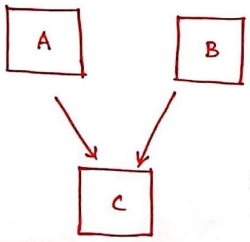
b) Multiple Inheritance :- (X)

More than one parent class is accessed by one child class.

This is not possible in Java. Because

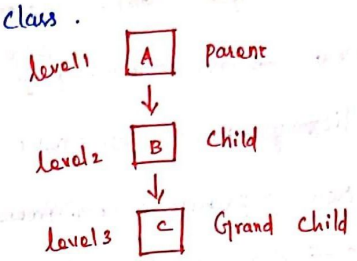
Reason : 1. There will be priority Issue.

2. It can Overcome by Interface.



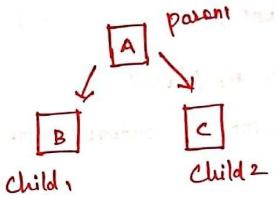
c) Multi-level Inheritance :-

One parent class is accessed by one child class which is accessed by another child class.



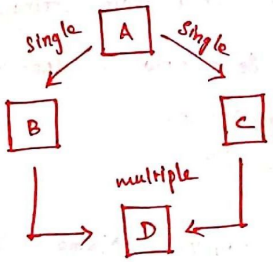
d) Hierarchical Inheritance :-

One parent class is accessed by more than one child class.



e) Hybrid Inheritance :-

Combination of Single & multiple Inheritance.



14/04/2019

Java Installation :-

Using Eclipse tool.

Creating class in Eclipse tool :- (open Eclipse tool)

File → New → java project → Give Project Name → Ok.

JRE System library → Src => Create package.

package : File → New → Package → com.greens.org → Ok

Class : File → New → Class → Give class Name → Ok

Basic Java program :-

```
package com.greens.org;
Public class Student {
    Public void StudentName() {
        System.out.println("Student Name Sudha");
    }
    Public void StudentId() {
        System.out.println("Student Id is 1000");
    }
    Public Static void main(String[] args) {
        Student St = new Student();
        St.StudentName();
        St.StudentId();
    }
}
```

o/p :- Student Name Sudha
Studan Id is 1000

(2) Encapsulation :-

a) Structure of creating folders.

b) Here code and data combined together as a single unit or entity.

(3) Abstraction :-

Hiding the implementation part.

It has 2 types.

1. Abstract class (partial abstraction)
2. Interface (fully Abstraction)

a) Abstract Class :

It Supports both the abstract method and non-abstract method.

There is no implementation part.

There will be only method name (or) Signature

We cannot create object for abstract class. Because there is no implementation part.

Using "Extends" Keyword we can access abstract class.

Public abstract => This Keyword is mandatory.

*. Check ☑ abstract class while creating class.

Program for Abstract class :-

```
Package . com.greens.org ;
Public abstract class Student {
Abstract method { Public abstract void StudentName();
                  Public abstract void studentMark();
Non-abstract method { public void StudentId() {
        Sysrem.out.println (" Student Id is 1000 ");
    }
}
```

Accessing abstract class using "Extends" Keyword :-

```
Package com.greens.org ;
Public class School extends Student {
  @ Override
  public void StudentName () {
     Sysrem.out.println (" Student Name is Sudha ");
  }
   @ Override
     public void StudentMark() {
        Sysrem.out.println (" Student Mark is 90 ");
  }
    Public Static void main (String[] args) {
     School Sc = new School();
     Sc.StudentName();
     Sc.StudentId();
     Sc.StudentMark();
  }
}
```

b) Interface :- (Fully Abstraction)

It Supports only abstract method.

There is no implementation part.

We cannot create object for interface because there is no implementation part.

Using "Implements" keyword we can access Interface class.

Public Abstract => is default. No need to mention the keyword.

Program for Interface Class :-

```
Package com.greens.org ;
public interface StudentInt {
    void StudentId() ;
    void StudentName();
    void StudentMark();
}
```

Oracle login

r.n.sudhamathi@gmail.com

Yurish @11

Accessing Interface class using "Implements" :-

```
Package com.greens.org;
Public class School implements Student Int{
 @ override
   public void StudentId() {
      System.out.println("100");
 }
 @ override
   public void Student Name();
      System.out.println(" Sudha ");
}
@ Override
   public void Student Mark() {
      System.out.println("60");
 }
   public static void main (String[] args) {
     School sc = new School();
     sc.Student Id();
     sc.Student Name();
     sc.Student Mark();
 }
}
```

20/4/2019 https://login.oracle.com/mysso/signon.jsp

Java Installation :- Download jdk (Java folder).

c: program files → program files → jdk & jre (1.8)

computer → right click → prop → control panel → System & security → Advanced system Settings → Advanced → Environment Variables → User Variable → New → Variable Name: JAVA_HOME

variable Value: (c: prog files /Java/jdk 1.8.0_181).
(paste jdk path)

System variables → path → open → Edit → New variable value → put ; paste bin path.
(paste bin path)

Run → cmd → java -version

Eclipse free download → download Eclipse

21/04/2019

Polymorphism :-

There is 2 types of polymorphism. poly means "Many". Morphism means "Ways or form". One task can be completed by many ways.

Types :-

1. Method overloading :-

```
Package com.greens.org;
public class Employee {
```

Class Name is Same.

Method Name Same.

Arguments or parameters are different.

Argument will be differ in 3 ways.

1. Datatype
2. Datatype count
3. Datatype model. order.

```
package com.greens.org;
public class Employee {
    public void employee (int id, char grade) {
        System.out.println ("Emp id & grade are "
                + id + ", " + grade);
    }
    public void employee (char grade, int id) {
        System.out.println ("Emp grade & id are "+grade+", "+id);
    }
    public void employee (String name) {
        System.out.println ("Emp Name is "+name);
    }
    public static void main (String[] args) {
        Employee e = new Employee();
        e.employee ("Arun");
        e.employee (100, 'A');
        e.employee ('B', 111);
    }
}
```

o/p :-

Emp Name is Arun
Emp id is 100.

2. Method Overriding :-

Class Name different.

Method Name Same.

Arguments (or) parameters are Same.

Super class program :-

```
package com.greens.org;
public class Boy {
  public void girlName()
    Sysrem.out.println ("Sudha");
  }
}
```

program for overriding :-

```
package com.greens.org;
public class Marriage extends Boy {
  @override
    public void girlName()
      Super.girlName();
  }
  public static void main (String[] args){
    Marriage m = new Marriage();
    m.girlName();
  }
}
```

o/p :-

Sudha.

Upcasting :-

Assigning object of the child as into parent class. This is "possible" in java.

```
Boy m = new Marrigage();  ✓ possible.
m.girlName();
```

Down casting :-

Assigning the object of parent class into Child class. This is "not possible" in java.

```
Marriage m = new Boy();  ✗ -Not possible.
m.girlName();
```

Access Specifier :-

4 types.

1. public
2. private
3. protected
4. default.

public : Global level access.

Inside and Outside the package.

private : Class level Access.

Can access only inside the class.

protected :

Inside and outside the package but we have to use extend keyword.

Default : package level Access.

Inside the package.

Scanner Class :-

Getting the input from user.

Scanner data types :-

nextByte()
nextShort()
nextInt()
etc...

next() => used for String for one word (or) Single word.

nextLine() => used for more than one word (or) Sentence.

Syntax

```
Scanner sc = new Scanner(System.in);
```

sc ↓ Scanner object

Default package name for Scanner :-

java.util.Scanner

Return Str type :-

control + 2 , L

Program for Scanner :-

```
Package com.geeks.org
import java.util.Scanner;
Public class S {
  Public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a number");
    int number = sc.nextInt();
    System.out.println("you have entered "+number);

    System.out.println("Enter a word");
    String word = sc.next();
    System.out.println("you have entered "+ word);

    sc.nextLine();
    System.out.println("Enter a sentence");
    String sentance = sc.nextLine();
    System.out.println("you have entered "+sentance
  }
}
```

O/P :-

```
Enter a number
87
You have entered 87
Enter a word
Yuvi
You have entered Yuvi
Enter a Sentance
Yuvi is my son
you have entered Yuvi is my son.
```

28/04/2019

Constructor :-

Class name and Method name Should be same.

It will not have return type (like void, int etc)

We do not need to call the constructor. It will execute automatically when we create the object.

It will Support method Overloading nor over riding.

Types :-

2 types of constructor.

1. Default constructor (Non parameterized constructor)
2. parameterized constructor (Should pass the argument)

Program for Constructor :-

```
package com.greens.org;
Public class Student{
default constructor  Public Student(){
                       Syscem.out.println("I am default constructor");
                     }
parameterized constructor  Public Student(int a){
                             Syscem.out.println("Student id is "+a);
                           }
   public Static void main(String[] args){
      Student st = new Student();  => calling default const
   }  Student st = new Student(2000); => calling parameter constructor
}
```

Constructor Chaining :-

```
package com.greens.org;
public class Student{
   public Student(){
      this(100);
    Syscem.out.println("I am default constructor");
   }
   public Student(int a){
    this("Sudha");
    Syscem.out.println("Student id is "+a);
   }
   public Student(String name){
    this('A');
    Syscem.out.println("Student name is "+name);
   }
   public Student(Char grade){
    Syscem.out.println("Student grade is "+grade);
   }
   public Static void main(String[] args){
     Student st = new Student();
   }
}
```

Control Statement :-

For
While
do while

While :- Entry level checking.

program for control statement :-

```
package com.greens.org;
Public class A {
  Public static void main (String[] args){
    int i=1;
    while (i<10){
      System.out.println(i);
      i++;
    }
  }
}
```

o/p :- 1 to 9.

Do while :-

Exit level condition checking.

```
package com.greens.org;
public class A {
  public static void main (String[] args) {
    int i=1;
    do
    { System.out.println(i);
      i++;
    } while (i<10);
}
```

For :- i=0 (initialization) ; i<=3 (condition) ; i++ (iteration)

```
package com.greens.org;
public class A {
  public static void main (String[] args) {
    int i;
    for( i=1; i<=10; i++) {
      System.out.println (i);
                (or)
      System.out.print (i)  => Same line o/p
      System.out.print(i+",") => with ,
}
}
}
```

o/p :- 1
2
3
⋮
10

(or)

```
int i;
for (i = 10; i>=1; i--) {
  System.out.println (i);
          (or)
  print (i+" ");
```

o/p :-
10 9 8 7 6 5 ...

```
1
22
333

3
22
111

123
12
1

1
12
123

*
**
* * *

* * *
**
*

1 1 1
2 2 2
3 3 3
```

o/p:- 1 2 3
1 2 3
1 2 3

```
Package com.greens.org;
Public class A {
    Public Static void main (String[] args){
     int i,j;
     for (i=1; j<=3; i++){
       for(j=1; j<=3; j++) {
          System.out.print(j);
       }
       System.out.println();
    }
  }
}
```

o/p:- 1
1 2
1 2 3

```
package com.greens.org;
public class A1 {
  public Static void main (String[] args) {
    int i,j;
     for (i=1; i<=3; i++) {
        for (j=1; j<=i; j++) {
          System.out.print("*");
     }
      System.out.println();
   }
  }
}
```

o/p:- 1 2 3
1 2
1

```
Package com.greens.org;
public class A2 {
   Public Static void main (String[] args) {
      int i,j;
      for (i=3; i>0; i--) {
         for (j=1; j<=i; j++) {
            System.out.print(j);
         }
         System.out.println();
      }
   }
}
```

o/p:- *
* *
* * *

```
Package com.greens.org;
Public class A3 {
   public Static void main (String[] args) {
      int i,j;
      for (i=1; i<=3; i++) {
         for (j=1; j<=i; j++) {
            System.out.print("*");
         }
         System.out.println();
      }
   }
}
```

o/p :-
```
* * *
* *
*
```

```
package com.greens.org;
public class A4 {
    public static void main(String[] args) {
        int i, j;
        for (i=3; i>0; i--) {
            for (j=1; j<=i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

o/p :-
```
1
2 2
3 3 3
```

```
package com.greens.org;
public class A5 {
    public static void main(String[] args) {
        int i, j;
        for (i=1; i<=3; i++) {
            for (j=1; j<=i; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

o/p :-
```
1 1 1
2 2 2
3 3 3
```

```
package com.greens.org;
public class A6 {
    public static void main(String[] args) {
        int i, j;
        for (i=1; i<=3; i++) {
            for (j=1; j<=3; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

o/p :-
```
3
2 2
1 1 1
```

```
package com.greens.org;
public class A7 {
    public static void main(String[] args) {
        int i, j;
        for (i=3; i>0; i--) {
            for (j=3; j>=i; j--) {
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```

```
3
for (i=3; i>=1; i--)
{
for (j=3; j>=i; j--)
  {
    s.o.p(i);
  }
s.o.p ln();
}.
```

04/05/2019

Array :- It Stores multiple value in a Single variable.

Similar

It Supports only ~~Single~~ datatype.

It is index based.

Index Starts from "0 to n-1".

n = Length.

```
Package com.greens.org;
Public class Arrays(){
    Public Static void main(String[] args){
        int a1 = 10;
        int a[] = new int[5];
    }
}
```

Returns to a multiple values to a single variable.

*. Adv :-

1. If any index is not assigned default value will be printed. (default value = 0).
2. If you override the value it will take the last value.

Dis Adv :-

It Supports only Similar datatype.

It has fixed length. (Run time error)

It has high level memory wastage.

```
        int a1 = 10;
// Syntax of array
int a[] = new int[5];
a[0] = 10;
a[1] = 20;
a[2] = 30;
a[3] = 40;
a[4] = 50;
System.out.println(a1);
int length = a.length;
System.out.println(length);
for (int i=0; i<a.length; i++){
    System.out.println(a[i]);
}
}
}
// enhanced for loop or foreach.
for (int x:a){
    System.out.println(x);
}
}
}
```

O/P :-
10
5
10
20
30
40
50

O/P :-
10
20
30
40
50

It prints all the values.

String :-

collection of words or characters in the double quotes.

String Methods

length()
equals()
equalsignorecase()
contains()
.Splited()
touppercase()
tolowercase()
.substring()
indexof()
lastindexof()
replace()
Startswith()
endswith()
isempty()
compareto()

Ctrl Space - Suggestion
Ctrl+D - Deleting a single line.
Ctrl2+L - Getting return type
// Command
/* ... */ - multiple lines command.

```
package com.greens.org;
public class StringManipulation {
    public static void main(String[] args){
        String s= "Welcome to Java";
```

length : `int le =s.length();` `Sysram.out.println(le);` 5

equals : `boolean equals = s.equals("welcome to Java");` `Sysram.out.println(equals);` false

equals ignore case : `boolean equalsIgnoreCase = s.equalsIgnoreCase("welcome to Java");` `Sysram.out.println(equalsIgnoreCase);` true.
bp

contains : `boolean contains = s.contains("wel");` `Sys.out.println(contains);` true

uppercase : `String upperCase = s.toUpperCase();` `Sys.out.println(upperCase);` WELCOME TO JAVA

lowercase : `String lowerCase = s.toLowerCase();` `Sysr.out.println(lowercase);` welcome to java

indexOf : `int indexOf = s.indexOf("w");` 0
`s.indexOf("e");` 1
`s.indexOf("x");` -1 default =0 already index exists so it prints -1
`Sys.out.println(indexOf);`

`int lastIndexOf = s.lastIndexOf("e");` `Sys.out.println(lastIndexOf);` 6

replace : `String replace = s.replace("welcome", "AAAA");` `Sys.out.println(replace);` AAAA to Java

Startswith : `boolean startsWith = S.startsWith("wel");` `Sys.out.println(startswith);` True

endswith: boolean endsWith = s.endsWith("va");
Sys.out.println(endswith); True

empty: boolean empty = s.isEmpty();
Sys.out.println(empty); False

subString:- String subString = s.subString(4); ome to Java
Sys.out.println(subString);

String subString2 = s.subString(4,9); ome t
Sys.out.println(subString2);

Split :- String[] split = s.split(" ",2); welcome
for(i=0; i<split.length; i++){ to Java
Sys.out.println(split[i]);

s.split(" "); welcome / to / Java

s.split("e"); w / lcom / to Java

s.split("e",2); w / lcome to Java

}

}

compare to() :-

Ascii value (A-Z)

65-90 (A-Z)

97-122 (a-z)

0-9 (48 to 57)

Remaining are Special characters.

// based on Ascii value.

String s1 = "A"; 65 o/p -1

String s2 = "B"; 66

int compareTo = s1.compareTo(s2);

Sys.out.println(compareTo);

}

}

String s1 = "c"; o/p 1

String s2 = "B";

* If you have many character comparisons based on first differing character.

String s1 = "ABCE"; o/p 1

String s2 = "ABCD";

* If different length same occurrence Then the comparison is based on length.

String s1 = "ABCDEFGH";

String s2 = "ABCD"; o/p :- 4 (4 letters extra)

* If different length and different occurrence then the comparison is based on first differing character.

String s1 = "ABCDEFGH"; o/p :-

String s2 = "ABCDF"; -1

11/05/2019

Collections :- To overcome all the disadvantages of array.

It Supports dissimilar datatype.

Dynamic memory allocation.

No memory wastage.

Types :-

Inter face
1. List => Array list, Linked List, Vector List } class
2. Set => Hashset, Linked Hashset, Treeset - class
3. Map => Hashmap, Linked HashMap, TreeMap, Hash Table, concurrent Hashmap - class.

① List :-

List is an interface. It prints in insertion order. It is Index based. It Allows duplicate value.

```
public
package com.greens.org;
import java.util.ArrayList;
import java.util.List;
public class Student{
  public static void main(String[] args){
    List li = new ArrayList();
    li.add(10);
    li.add(100.12);
    li.add(true);
    li.add('s');
    li.add("Hello");
    System.out.println(li);
// to find the length of the list
    int size = li.size();
    Sys.out.println(size);
// to print particular value
    Object object = li.get(1);
    Sys.out.println(object);
// to iterate
    for(int i=0; i<size; i++){
      Sys.out.println(li.get(i));
    }
    for(Object x : li){
      Sys.out.println(x);
    }
  }
}
```

o/p :-

```
[10, 100.12, true, S, Hello]
5
100.12
10
100.12
true
S
Hello
10
100.12
true
S
Hello.
```

Generics :-

It supports particular datatype.

From Java 1.6 feature < it is available.

prino in insertion order.

Allows duplicates.

```
List <Integer> li = new ArrayList <Integer> ();
```
(Generics)

```
li.add (20);
li.add (10);
li.add (30);
li.add (60);
li.add (10);
li.add (10);
li.add (null);
li.add (null);
li.add (40);
Li.add (50);
li.add (80);
li.add (70);

Sys.out.println (li);

int size = li.size();
Sys.out.println (size);

Integer integer = li.get (3);
  Sys.out.println (integer);
for (int i=0; i<size ; i++){
   Sys.out.println (li.get(i));
}
}
```

o/p:-

[20, 10, 30, 60, 10, null, null, 40, 50, 80, 70]

```
12
60
20
10
30
60
10
10
null
null
40
50
80
70
```

Remove :-

// to remove particular value by passing its index.

// if we remove the value, Index order will not change.

// Index value will move ~~backward~~ forward. (backward)

```
li.remove (2);
Sys.out.println (li);
```

[20, 10, 60, 10, null, null, 40, 50, 80, 70].

Add :-

// Index based add

// If we add a value to particular Index, Index order will not change.

// Index value will move ~~forward~~. backward. (forward)

```
li.add (2, 1000);
System.out.println (li);
```

[20, 10, 1000, 60, 10, 10, null, null, 40, 50, 80, 70]

Replace :-

Set is a method to replace.

```
li.set (3, 1000);
System.out.println (li);
```

[20, 10, 30, 1000, 10, 10, null, null, 40, 50, 80, 70]

Index of :-

To print index of particular value,

If value is available, it will print the relevant index position.

If value is not available, it will print -1

If multiple value is available, it will print first index point.

```
li. indexOf = li.indexOf (30);      2

li. indexOf = li.indexOf (300);     -1

li. indexOf = li. indexOf (10);     0 (it will print first index point).
Sys. out. println (indexOf);
```

lastIndexOf :-

If multiple values are available, it will print last index position.

```
int lastIndexOf = li. lastIndexOf (10);
sys. out. println ( lastIndexOf );      5
```

To print all the index of 10.

```
for (i=0; i<size; i++) {
  if (li.get(i)==10) {
    Sys.out.println(i);
```

1
4
5

contains :-

```
boolean contains = li. contains (10);      it.
Syst em. out. println (contain);           true.
```

clear :-

method to clear all the list.

```
li. clear();
Sys. out. println(li);                     [ ]
```

isEmpty :- To check whether the list is empty or not.

```
boolean empty = li. isEmpty();             true.
Sys. out. println (empty);
```

addAll :-

method to copy to list 1 to list 2.

```
List <Integer> li2 = new ArrayList<>();
li2. addAll (li);                          [20,10,30,60,10,10, 40,50,80,70]
Sys. out. println (li2);                   [20,10,30,60,10,10,40 50,80,70]

List li2 = new ArrayList<>();
li2.addAll(li);                            [20,10,30,60,10,10,40,50,80,
li2. add (1000);                           [20,10,30,60,10,10,40,50,80,7
li2. add (2000);                                1000,2000,3000]
li2. add (3000);
```

Remove all ;-

To remove list1 values in list2

list2 - list1

```
li2.removeAll(li);          [1000, 2000, 3000]
Sy.out.pr(li2);
```

Retain All :-

To compare two list and print the common values.

```
li2.retainAll(li);          [20,10,30,60,10,10,40,
Sys.out.println(li2);        50,80,70].
```

Linked list methods :- Node based.

As same as array list methods.

Diddadv Searching & retrieving is difficult.

Adv :- Adding & deleting is easy, bcoz it is seperate node, if we want to search 100th index, it starts from 0th index & retrieve the value.

Adv of Array list :- Series based. (best case)

Searching & retrieving is easy.

Diddadv of Arraylist :- (worst case)

Adding & deleting bcoz if we remove/ add values, nex index values will move forward/ backward.

Vector List :- execution.

Synchronized and Thread Safe (It allows only one by one process).

Array List :-

Asynchronized and not thread safe (parallel).

② Set :- It is Interface.
It is value based.

Print random values. (not maintain any order).
It will allow single 'Null' value but not duplicate Null
Does not allow duplicates.
We cannot use normal for loop. we can use only enhanced for loop. bcoz it is not index based.

```
import java.util.ArrayList;
        HashSet;          Empty - memory will be created but empty set.
        LinkedList;       Null
        List;             No memory will be
        Set;              created for null
        Vector;
public class s {
    public static
Set<Integer> x = new HashSet<>();
 x.add(10);
 x.add(20);
 x.add(30);
 x.add(40);
 x.add(10);
 x.add(60);
 x.add(60);                          [20,40,10,60,30]
  Sys.out.println(x);
int size = x.size();                 5
Sys.out.println(size);
 x.remove(30);                       [20,40,10,60]
```

Hashset :-

Random order

It allow single null, not duplicate null.

Linked Hash set :-

Insertion order.

Allow single null.

Tree set :-

Ascending order → Based on ASCII value.

Not allow even single null value. because it is based on Ascii value. There is no Ascii value for null.

To avoid duplicate values we can assign the List to set.

```
List <Integer> x = new ArrayList<>();
x.add (20);
x.add (30);
x.add(10);
x.add (40);
x.add (10);
x.add(60);
x.add(60);
x.add(40);
x.add(70);
Sys.out.pr(x);        [20, 30, 10, 40, 10, 60, 60, 40, 70]
Set <Integer> y = new HashSet<>();
y.addAll(x);
Sys.out.pr(y);        [20, 70, 40, 10, 60, 30]
```

Array list - set list => duplicate count.

```
int duplicateCount = x.size() - y.size();        3.
Sys.out.pr (duplicateCount);
```

(3) Map :-

It is an key and value pair.

One key and one value → entry.

key will not allow duplicates. but value will allow duplicates.

"put" is the method to add value.

```
Map <Integer, String> x = new HashMap<>();
// to add value - use method "put"
x.put (1, "JAVA");
x.put (2, "Selenium");
x.put (3, "Cucumber");        {1=JAVA, 2=Selenium, 3=Cucumber}
Sys.out.pr (x);
// to find length
int size = x.size();
Sys.out.pr (size);        3
// to print particular value
String string = x.get(3);
Sys.out.println (string);        Cucumber.
// to print only keys.
Set <Integer> Keyset = x.keySet();
Sys.out.println (Keyset);        [1, 2, 3]
// to print only values.
Collection <String> values = x.values();
Sys.out.pr (values);        [JAVA, Selenium, Cucumber]
```

// to iterate

```
Set<Entry<Integer, String>> entrySet = x.entrySet();
for(Entry<Integer, String> z : entrySet){
    Sys.out.println(z);
```

1 = JAVA, 2 = selenium, 3 = Cucumber.

```
Sys.out.println("Key value is"+z.getKey+"value is"+
        z.getValues());
```

Key is 1 value is Java

Hash Map :-

Random order

Key => No duplicate

Value => Allows "

Key => Allow Single null

value => Allows multiple null.

Linked Hash Map :-

Insertion order.

Key :- No dup, Allow single null, Value :- Allow dup, Allows multiple null.

Tree Map :-

Ascending order. (Based on key).

Key - Not allows even single null becoz ascending order is based on - Ascii value.

Value - Allow null.

HashTable :-

Random order.

Key and Value will not even allow single null.

Concurrent Hash map :

Synchronized, Thread safe (one by one).

Hash Map :

Asynchronized, Not thread safe (parrallel process).

12/05/2019

User defined Map :-

It is an example of

POJO : plain old Java Object / Bean class / Model class.

If you give private to the variable, it can be accessed indirectly using getters and Setters.

Right click → Source → Generate getters and Setters → select All → Generate.

```
package
pub class Student {
    private int StudentId;
    private String StudentName;
    public int getStudentId() {
        return StudentId;
    }
} public void setStudentId(int StudentId){
        this.studentId = StudentId; }
public String getStudentName(){ return StudentName;}
public void setStudentName(String StudentName){
        this.StudentName = StudentName) } }
```

```
package
    util
public class UserDefineMap {
    public static void main (String[] args) {
    Map < String, Student > x = new HashMap<>();
                  ↓
          user defined datatype.
          class is also a datatype so here we are using
                                          class name.
    Student S1 = new Student(); => Assigning S1 object
                                   to map (Student).
     S1 . setStudentId (1);
     S1 . setStudentName ("ABC");

     Student S2 = new Student();
      S2 . setStudentId (2);
      S2 . setStudentName ("xyz");

      x.put ("G1", S1);
      x.put ("G2", S2);
  Set < Entry <String, Student>> entrySet = X.entrySet();
  for ( Entry <String, Student> entry : entrySet) {
    Sys.out.pr (entry); => It will print only reference
  Sys.out.pr ("college id is = "+ entry.getKey() + "and
      Student id is = " + entry.getValue().getStudentId()+
     "and Student name is = "+entry.getvalue().
                          getStudentName());
  }
}
}
```

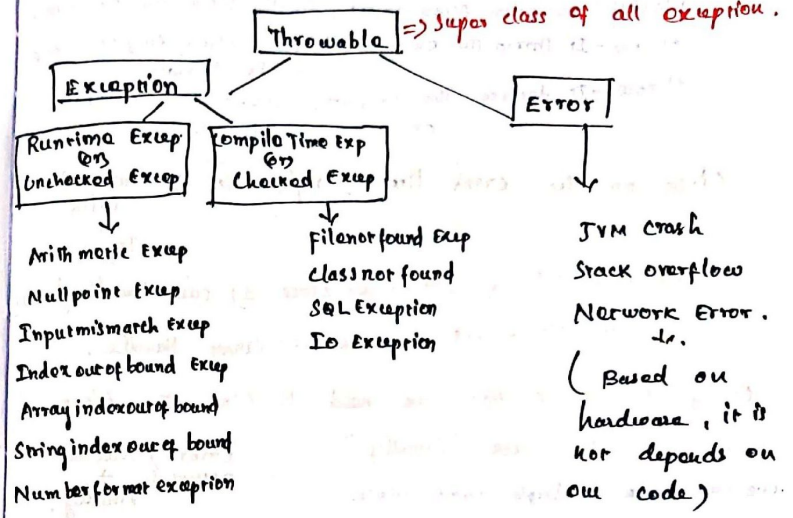
O/P :-

college id is = G1 and Student id is = 1 and Student name is = ABC.

college id is = G2 and Student id is = 2 and Student name is = xyz.

Exceptions :-

Exception is like a error, if exception occurs programs will terminate at that line itself.



To convert String to Num

```
String s = "089435648";
int parseInt = Integer.parseInt(s);
Sys.out.pr (parseInt);
```

O/P

089435648

To convert Num to String

```
String valueOf = String.valueOf (parseInt);
Sys.out.pr( valueOf );
```

Exception Handling :-

Try - It will throw the exception.
catch - It will catch the exception.
finally - Exception occurs or not, finally block will execute.
throw - It throws the exep, inside the method, only one exep can be thrown.
throws - It declare the exception, method level, multiple exception can be declared.

Click on the error line => right click -> Surround with + Try -> catch

Sub class 1st => 2nd super class => can Handle/parse

Super class 1st -> 2nd sub class => cannot Handle.

Excep occurs or not we need to close the block

So we will use "finally"

we can have multiple catch block

error occurs } catch + finally.

No error => Finally.

```
main
    S.o.p(1);
    S.o.p(2);
    try {
No error String s = null;
        try {
error but no catch  S.o.p(3/0);
        }
        finally {
            S.o.p("Inner finally");
        }
    } catch (Exception e) {
            S.o.p("Outer Catch");
    }
    Finally {S.o.p("Outer Catch finally");
    }
    S.o.p(4);
}
```

o/p :-
1
2
Inner finally
Outer catch
Outer finally
4.

```
S.o.p(1);
S.o.p(2);
try {
    String s = null;
    try {
        try {
            S.o.p(3/0);
        }
        Catch (Exception e) {
            S.o.p("Nested Inner Finally");
        }
        S.o.p(3/0);
    }
    finally {
        S.o.p("Inner finally");
    }
} catch (Exception e) {
    S.o.p("Outer Catch");
} finally {
    S.o.p("Outer Finally");
}
S.o.p(4);
}
}
```

One try -> one finally
multiple catch.

o/p :-
1
2
Nested Inner Catch
Nested Inner Finally
Inner finally
Outer Catch
Outer finally
4

If it is "Exception" it will shows only exception in o/p not print the msg.

```
public class UserDefineMap {
    public static void info() throws Exception {
        throw new StudentDetailNotFound();
    }
    public static void main (String[] args) throws ArithmeticException,
                NullPointerException, Exception {
        info();
    }
}
```

user defined exception. It will print the msg as well

```
package
public class StudentDetailsNotFound extends Exception {
    public StudentDetailsNotFound() {
        Sys.out.println ("Student details not found");
        getmessage();
    }
}
```

o/p :- Exception in thread "main" Student details not found

12/05/19

Types of variables :-

3 Types.

local Variable
Class Variable
Static Variable.

local variable :-

Inside the method, has to be declared. Have to initialize the value. we can use only inside the method.

Class Variable :-

Inside the class & outside the method has to be declared.

No need to initialize the value.

If we did not initialize the value it will take default value.

It is an "Instance Variable." We can use inside the class anywhere.

we should call the variable by using instance.variable. (ie) obj.variable.

Static Variable :-

We can call the variable without object (or) instance.

Against object.

we should use the word "static" while declaring the Static Variable.

we can access the Static Variable inside and outside the method.

Package

```
Public class A {
    Static int b=100;
    public void empId() {
        int a=10;
        s.o.pln ("The value of a" +a);
        s.o.pln (" The value of b"+b);
    }
    public Sratic void main
        A obj = new A();
        obj. empId();
        S.o.pln (b);
    }
}
```

O/P:-

The value of i 10
The value of a 100
100.

We should use "public" keyword to extend the variable in another class.

Static

final

1. We can give the keyword in
   before a. Variable
   b. method
   c. Static block.

2. We can call Static method without creating Object.

3.

final :-

We can use the keyword infront of

a. class
b. Method
c. Variable.

final class :-

Cannot be inherited.

final method :-

We cannor override the method in another class.

final Variable :-

We cannor change the value of the variable.

Break :-

It will terminate or exit from the loop.

Continue :-

It will skip the particular iteration.

If // if - else if - else block.

else,

```
String empName = "AAA";
int empId = 10;
if (empName == "AAA") {
  S.o.pln ("Emp Grade is A");
}
else if (empId == "10") {
  S.o.pln ("Emp Grade is B");
}
  else
    S.o.pln ("Emp Name is Invalid");
```

&& => Both conditions should match. logical AND.

|| => logical OR. Any one condition should match.

```
if (empName == "AAA" || empId == 20){
    S.o.pln ("Emp grade is A"); }.
```

o/p :- Emp grade is A.

```
if (empName == "AAA" && empId == 20){
    S.o.pln ("Emp grade is A"); }
```

o/p :- Emp Name is Invalid.

Switch :-

```
switch (key) {
  case value:
    break;
  default :
    break; }
```

It will print the value, when the value matches with it.

```
package
    main
    int id = 2;
    Switch (id) {
    case 1 :
        S.o.pln (" Student Name is Sudha");
        break;
    case 2 ;
        S.o.pln (" Student Name is Sathya");
        break;
    default :
        S.o.pln (" Student Id is Invalid");
        break;
    }
```

this :- To access class variable.

Super :- To access the parent variable.

```
package green;
public class B {
    int a = 1000;
}
```

```
package greens;
public class A extends B {
    int a = 100;
  public void empId() {
     int a=10;
     s.o.pln (" The value of local variable " + a);
     s.o.pln (" The value of class variable '+ this.a);
     s.o.pln ( "The value of parent variable " +super.a);
  }
   public static void main ( String [] args ) {
         A  a = new A();
        a. empId();
   }
```

Web driver ⇒ is a library file (jar file). is a predefined files. we can au

Selenium:-

It is a tool to automate the web applications.

jar :- predefined classes & files.

Selenium

project → New → give proj name → create new folder Drivers → create new folder Library.

Check version of the browser. Settings → Help → about Chrome.

Google → Selenium download → Click on 1st link → copy the link and paste in notepad. https://www.seleniumhq.org/download/ . ⇒ Download version → (3.141.59) click on this.

download Selenium → Click (3.141.59)

Mozilla → mozilla gecko driver → https://github.com/mozilla → firefox → mozilla Gecko driver.

IE → IE Driver server.

copy jar file → Eclipse → paste in library.

copy Chrome driver → eclipse → paste in Drivers. IE & Mozilla.

Src → right click → create package → com.selenium.day1 → create class.

Add to build path.

main → rc → build path → ~~configure~~.
on selenium jar file.

WebDriver → Interface

driver → object. Reference.

ChromeDriver → Class.

To maximize the browser:

```
driver.manage().window().maximize();
```

To get page title:

```
String title = driver.getTitle();
S.o.pln(title);
```

To get current url:

```
String currentUrl = driver.getCurrentUrl();
S.o.pln(currentUrl);
```

o/p :-

maximize => webpage will be maximized.

title => Facebook - log in or sign up

currenturl => https://www.facebook.com/

To give wait time :-

```
Thread.sleep(10000);
```

To navigate to url :-

```
driver.get("https://www.facebook.com");
```

To close the browser :-

close :- closes only current browser.

quit :- closes all the opened browsers.

```
driver.close()
driver.quit()
```

To Navigate the Browser :-

```
driver.get("https://www.google.com");
driver.navigate().to("https://facebook.com");
driver.navigate().back();
driver.navigate().forward();
driver.navigate().refresh();
```

19/05/2019.

By Locators :- (8 types)

1. id
2. name
3. Class name
4. X path
5. CSS Selector
6. tag name
7. link text
8. partial link text

preference

id
name
x path
CSS

Inspect :-

tr/td/ - html tags. (Pink)

Class - Attribute name (orange)

Blue color - Attribute value.

To find element / locator :-

find element -(By locators)

To find the locator / to find the web element. [findElement(By. name / id / xpath)].

TO input values to webpage :-

SendKeys()

TO click button :-

Click()

id
name
class name
} direct.

If ID changes each time we have to write x path with other attribute Name.

Other things we have to write the x path.

xpath 1

(1) //tagname [@attributeName = 'AttributeValue']

//input [@value = 'Log In'].

write x path → go to DOM Structure →

Ctrl + F => If it finds the value then the x path is correct. Otherwise xpath is wrong.

If there is no id, name or x path we need to use following xpath

(2) //tagname [text() = 'value']

xpath 2

getText() => It is a method to get text value from the website.

WebElement => It is the return type.

25/05/2019

Types of x path :- (5 types)

Refer 1 & 2 above.

3. Webelement changing dynamically :-

//tagname [contains(@attributename, attributevalue)]

4. //input [contains(text(), 'value')]

5. Based on Index :-

(//input [@attributename = 'attribute value'])[i].

26/05/2019

Drop Down

Scroll

Drag & drop

Drop down :-

Select - Tag name.

3 types :-

Index (int)
Value
Visible text } String.

```
package
class { public static void chromeBrowserLaunch() throws InterruptedException {
Sys.property("key", "value");
WebDriver driver = new ChromeDriver();
driver.manage();
driver.get("https://www.facebook.com");
WebElement dayDropdown = driver.findElement(By.id("day"));
WebElement monthDropdown = driver.findElement(By.id("month"));
WebElement yearDropdown = driver.findElement(By.id("year"));
                                                    (same for next prog)
Select dd = new Select(dayDropdown);
dd.selectByIndex("10"); -> int
Select md = new Select(monthDropdown);
md.selectByValue("7"); -> String
Select yd = new Select(yearDropdown);
yd.selectByVisibleText("2018"); -> String
}
public static void main(String[] args) throws InterruptedException {
ChromeBrowserLaunch();
}
}
```

getFirstSelectedOption();

It is the method to see the value presented in the website initially & after changed the value.

```
Select md = new Select(monthDropdown);
WebElement firstSelectedOption = md.getFirstSelectedOption();
String text = firstSelectedOption.getText();
Sys.out.pln("Before Selection" + " " + text);
md.selectByValue("7");
WebElement selectedMonth = md.getFirstSelectedOption();
Sys.out.pln("After Selection" + " " + selectedMonth.getText());
```

o/p :-

Before Selection May

After Selection Jul.

getOptions() :-

It will print all the values in the dropdown.

List<WebElement>

```
List<WebElement> list = md.getOptions();
for (WebElement webElement : list) {
Sys.out.pln(webElement.getText());
}.
```

o/p :-

Month
Jan
Feb
Mar
Apr
May
June
Jul
Aug
Sep
Oct
Nov
Dec.

Ctrl + Shift + / => to comment - /* ... */

Multi Select List :-

Used to select more than one options.

```
driver.get
( https://www.seleniumeasy.com/test/basic-select-dropdown-demo.html )

WebElement multiselect = driver.findElement(By.id("multi-select"));

Select ms = new Select(multiselect);

boolean multiple = ms.isMultiple(); - to check whether it is multi select or not
                                      o/p:- True

Sys.out.println(multiple);

ms.selectByVisibleText("New Jersey");
ms.selectByValue("Florida");
ms.selectByIndex(5);
```

To print the selected options :-
getAllSelectedOptions()

```
List<WebElement> allSelectedOptions = ms.getAllSelectedOptions();

for(WebElement webElement : allSelectedOptions){

    Sys.out.println(webElement.getText());

}
```

o/p :-

Florida
New Jersey
Texas

} In website options order.

To deselect : deselectBy()

ms.deselectByIndex(5); => It will deselect Texas

ms.deselectAll(); => It will deselect all the options.

SCHEMAS

create Schema (facebook) -> Credentials (Alt+4) -> Create table ->

Apply -> Column Name | Datatype | PK | NN | UQ | B

Column Name	Datatype
UserId	INT
UserName	VARCHAR(30)
Userpassword	VARCHAR(45)

Apply -> Finish.

Tables -> users -> Enter details in the table.

UserId	UserName	Userpassword
1	Sudha	V123
2	Mani	S123
3	Manju	S123

(or) directly we Insert into query -> Apply.

sdbc
API :- It will create connection b/w database and java code. It is used to execute the Sql statement.

04/06/2019

JDBC Connections

To fetch values from database.

MySQL Server - open Source. (maven JDBC mysql dependency 1. mysql - connector - java; Jdbc mysql jar)

Create New project → (JDBC_Connections) →
Create New folder (Library → to put jar files) →
Go to google → mysql jdbc driver jar (type) → (1st link) connector/J 8.0.17
select os: platform independent → Click on zip
(download) → Go to downloads → unzip →
copy jar file → Go to eclipse → paste in Library
→ rc → build path → Add to build path

```
package
public class JDBC Connections {
    public static void main (String[] args) {
        // Step 1: Load the driver
        // Step 2: Setup connection
        // Step 3: Create Statement
        // Step 4: Execute query
        // Step 5: Result Set
        // Step 6: to get data
        // Step 7: Close connection.
```

```
public static void main(String[] args) throws ClassNotFoundException, SQLException {
    Step 1: Class.forName ("com.mysql.jdbc.Driver");   // another way to call the class
    Step 2: String userName = "root";
            String password = "root";
            String dbUrl = "jdbc:mysql://localhost:3306/facebook";   // jdbc: API, mysql: db, localhost: ipadd, 3306: port; sql APIs used to execute sql queries
            String query = "SELECT * FROM Credentials.users;";
    Connection con = DriverManager.getConnection(dbUrl, userName, password);
    Step 3: Statement stmt = con.createStatement();
    Step 4: ResultSet rs = stmt.executeQuery (query);
    Step 5: while (rs.next()) {   // to iterate
    Step 6: rs.getString
            String userid = rs.getString (1);   // column Index
            String userName = rs.getString (2);
    String userpass = rs.getString ("User Password");   // column Name.
            System.out.println (userid);
            s.o.p (UserName);
            sys.out.pln (userpass);
    }
    Step 7: con.close();
```

1. mysql prerequisites
   a. C++ 2010
   b. .Net framework

To Install workbench

Search MySQL INSTALLER (1st link) → microsoft windows platform Independent (8.0.17)
→ download zip file → 1. mysql Server 2. mysql workbench 3. connector
Run → MySQL Workbench → database → Connect to db →
give username and password → connect. (If it is 1st time pass)

We should store the result set in "List or Map" to use it in Selenium.

Key – userId (Integer).

Value – UserName, Userpassword (String).

```
Static Map
    main
{
 System.setproperty ("webdriver.chrome.driver", "exe path");
 WebDriver driver = new ChromeDriver();
 driver.get ("https://www.facebook.com");
 driver.findElement(By.id ("email")). sendkeys(
 Map<String, String> dataFromDB = getDataFromDBC);
    String CD = dataFromDB.get ("1");
    S.o.p (CD);
    String[] Split = CD.Split("=");
    driver. findElement (By.id ("email")). Sendkeys (split[0]);
    driver. findElement (By.id ("pass")). Sendkeys(split[1]);
 }
public static Map<String, String> getDataFromDB (). throws
                    ClassNotFoundExcep, SQLExcep{
  try {
      Map <String, String> mp = new HashMap<>();
      Step 1 to Step6
```

Step 6:

```
      while (rs.next()) {
      String userid
             UN
             userpass
      mp.put( userId, UN+"="+userpass );
  }
    con.close();
    return mp;
  }
  catch (Exception e) {
     e.printStackTrace();
     throw new RunTimeException();
  }
}
```

15/06/2019 MAVEN

1. Managing dependencies 2. creating default folder structure

Maven download → click on 1st link → apache-maven-3.6.1-bin.zip → copy → c: → create folder (app) → paste → unzip → bin.

Cmd → mvn -v

copy the path upto apache → before bin → Sys prop → Advanced → Env variable → (to access maven throughout the sys.) → MAVEN_HOME (sys var) (without bin) → paste → ok → path → par ; then paste the path (upto bin)

Eclipse

File → New → Maven project → click on maven proj → Next → Next → Group Id (package Name) → com.selenium.maven → Artifact (proj name) → Maven Selenium → OK.

pom.xml under Maven Selenium → Selenium java common Io Junit dependencies

<dependencies> We need to add the jar files.

Go to google → Selenium Java maven dependency → click on Maven Repository → click on Selenium Java → 3.141.59 → copy the Maven code → go to eclipse → pom.xml → paste the code under <dependencies>

(X.)

C: → users → Yavish → ".m2" → It will contain all the jar file.

type common Io → tap click on 2.6 version → (for screenshot.) copy the code paste it in dependencies.

1. It will download the jar file from Internet automatically even if we delete the m2 folder in C:
2. Maven will create maven default folder Structure. (src/main/java), (src/test/java).

Create a folder src/test/resource for Browser. Create one package → copy & paste all the drivers.

Implicit wait: (define only once after initiating the browser.) (To avoid the usage of Thread.sleep)

1. To perform the action once the element is launched.
2. We can define the maximum wait time. (Before throwing "No Such Element Exception", it will wait for)
3. If the element is not launched until the wait time it will throw Exception.
4. We can use this for all the "find element". (until the element got destroyed. It will destroyed when calling "driver.quit".

driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);

Explicit wait:

We will use WebDriverWait class.

Here we have the method called until

```
WebDriverWait wd = new WebDriverWait(driver,20);
wd.until(ExpectedCondition.visibilityOf(element));
```

Base class :-

1. Select option from Drop Down :-

```
public static void selectOptionFromDropDown(WebElement element,
        String value, String options) throws Exception {
try {
    waitForElementVisibility(element);
    Select se = new Select(element);
    if (options.equalsIgnoreCase("value")) {
        se.selectByValue(value);
    }
    else if (options.equalsIgnoreCase("visibleText")) {
        se.selectByVisibleText(value);
    }
    else if (options.equalsIgnoreCase("Index")) {
        se.selectByIndex(Integer.parseInt(value));
    }
    else {
        throw new Exception("Not a valid option");
    }
} catch (Exception e) {
    e.printStackTrace();
    throw new RuntimeException();
}
}
```

2. Scroll by using Elements :-

We want to use the interface called "JavascriptExecutor".

Driver is also a interface.

To call the Interface using Interface we will use typecasting.

```
public static void ScrollbyUsingElements(WebElement element) {
try {
    JavascriptExecutor js = (JavaScriptExecutor) driver;
    // typecasting -> assigning the interface to driver.
    // js.executeScript("arguments[0].scrollIntoView();",
                        element);
} catch (Exception e) {
    e.printStackTrace();
    throw new RuntimeException();
}
}
```

3. ScrollBy coordinates () {-

```
public static void scrollByCoordinates(int width, int height) {
try {
    JavascriptExecutor js = (JavascriptExecutor) driver;
    js.executeScript("window.scrollBy(200,500)");
    js.executeScript("window.scrollBy("+width+","+height+")");
} catch (Exception e) {
    e.printStackTrace();
    throw new RuntimeException();
}
}
```

4. Send Keys using Actions() :-

It is class. We need to create object.

```
public static void sendKeysUsingActions(WebElement element,
                                        String value) {
    try {
        Actions ac = new Actions(driver);
        ac.sendKeys(element, value).build().perform();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

5. Click Using Actions :-

```
public static void clickUsingActions(WebElement element) {
    try {
        waitforElementVisibility(sre);
        Actions ac = new Actions(driver);
        ac.click(element).build().perform();
    } catch.
```

6. Drag and Drop Using Actions :-

```
public static void dragAndDropusingActions(WebElement
                                        sre, WebElement tar) {
    try {
        waitforElementVisibility(sre);
        Actions ac = new Actions(driver);
        ac.dragAndDrop(sre, tar).build().perform();
    } catch (Exception e) {
        e.printStackTrace();
        throw new RuntimeException();
}}}
```

7. Take Screen shot () :- It is a file. o/p should be file.

We have an interface called "TakesScreenshot".

```
public static void takeScreenShotonthepage() throws IoException {
                                    (String fileName) throws IoException
    try {
        TakesScreenShot ts = (TakesScreenshot) driver;
        file temp = ts.getScreenShotAs(outputType.FILE);
        File des = new File(System.getProperty("user.dir")+"
            \\src\\test\\resource\\com\\selenium\\report\\screenshot
            \\"+fileName+".png");
        FileUtils.copyFile(temp, des);
    } catch (WebDriverException e) { e.printStackTrace(); }}
```

↓ changes dynamically so we pass it as argument

To store the Screen shot :-

Create a package under src/test/resource:
com.selenium.report.screenshot.

pom (page object model) :-

1. Create a package under Maven Selenium as com.selenium.pom

2. To reuse the element & maintain it.

3. If we have 10 users then we need to create 10 find element for all the users. If the element changes from email to phone then we need to change all the id as phone. So here we can't reuse the element and can't maintain it. Also this is not a best practice. So we are going to overcome this disadvantage we are going for "POM".

3. Also readable.

We are creating seperate class for all the pages. So that we can easily update the code wherever the changes is applicable.

4. Under package create three classes for login page, registerpage & Home page.

- Homepage.java
- Loginpage.java
- Registerpage.java.

1. Define Elements.
2. Create getters.
3. Create a constructor. (No need to initialize the driver) so we are using constructor.
4. Assign the parent driver to current class driver.
5. pagefactory.initElements(driver, this);
   - pagefactory → class
   - initElements → method
   - this → element

To initialize the elements.

Typecasting:- Assigning one interface to another Interface.

22/06/2019

Mouse Over :-

(like, click, double click)

For any mouse interaction we will use the class called Actions.

```
Actions ac = new Actions(driver);   => parameterized constructor
ac.moveToElement(helloSingIn).build().perform();
Thread.sleep(3000);
driver.findElement(By.xpath("//div[@id='nav-al-signin']//span").click();
```

So many sign in are available. so we should come from its parent id

Scroll :-

To Scroll we should use the interface called JavascriptExecutor. and method called executeScript.

```
js.executeScript("arguments[0].scrollIntoView();", badminton);
```

To scroll down & up :-

- false → down
- True → up.
- Nothing → upto the element

```
WebElement amazonlogo.driver.findElement(By.xpath(" "));
WebElement badminton = " " " ";
JavaScriptExecutor js = (JavaScriptExecutor) driver;
js.executeScript("argument[0].scrollIntoView();", badminton);
Thread.sleep(3000);
Js.executeScript("argument[0].scrollIntoView()", amazonlogo);
```

Scroll using pixels :-

width Height

```
js.executeScript("window.scrollTo(0,500)");
Thread.sleep(3000);
js.executeScript("window.scrollTo(0,-200);
Thread.sleep(3000);
```

webpage visible, whole body of wp

```
js.executeScript("window.scrollBy(0,document.body.scrollHeight)");
```

no sign means it will go down; the entire height of wp.

```
Thread.sleep(3000);
js.executeScript("window.scrollBy(0,-document.body.scrollHeight)");
```

- means it will go up

Scroll ⇒ scrollIntoView() ⇒ while By defining the direct element

scrollTo() ⇒ using pixels

scrollBy() ⇒ to scroll the entire window up & down

23/06/2019

Drag and Drop :-

https://www.guru99.com/drag-drop-selenium.html

" test/drag_drop.html.

Take the Xpath for the source & destination elements and perform the action.

```
WebElement amount = driver.findElement(By.xpath(" "));
WebElement placeHolder = driver.findElement(By.xpath(" "));

Actions ac = new Actions(driver);
ac.dragAndDrop(amount, placeHolder).build().perform();
```

To pass the double quotes inside a String is also called as escape character.

Escape Character :-

```
String s = "hello \"sudha\"";
s.o.pln(s);
```

o/p :- hello "Sudha"

Screen shot :-

First we want download a jar file called "Common IO" to copy the files.

lastest Version - 2.6 ⇒ download & unzip

paste the jar in Library → Add to build path

des

```
File src = new File("C:\Users\yuvish\eclipse\-workspace\Selenium\library\Screenshot.png");
```

→ the path where u want to store the file.

→ interface

```
TakesScreenshot ts = (TakesScreenshot) driver;
```

→ screenshot will store in temp folder

```
File temp = ts.getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(temp, des);
```

↓ method to copy the files

Right Click :-

www.google.com

Rc on Images open in new window.

Take the Xpath for images.

For Keyboard actions we are using the Class called "Robot" Class.

```
WebElement images = driver.findElement(By.xpath("//a[text()='Images']"));
Actions ac = new Actions(driver);
ac.contextClick(images).build().perform();
```
↑ for right click

for keyboard actions

```
Robot r = new Robot();
r.keyPress(KeyEvent.VK_DOWN);
r.keyRelease(KeyEvent.VK_DOWN);
r.keyPress(KeyEvent.VK_DOWN);
r.keyRelease(KeyEvent.VK_DOWN);
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
Thread.sleep(2000);
```
↑ virtual key

google - child window
google images - parent window

To switch from parent window to child window :-

```
String cid = driver.getWindowHandle();
```
child id

↳ It will give the current driver id

```
S.o.ph(cid);
```
To avoid duplicates

```
Set<String> pid = driver.getWindowHandles();
```
parent id

↳ It will give all the driver ids

```
S.o.ph(pid);
for(String x : pid){
    if(x.equals(cid)){
        driver.switchTo().window(x);
        driver.findElement(By.name("q")).sendKeys("Amazon", Keys.ENTER);
        driver.close();
    }
}
```
↑ To switch from one window to another window

↓ To press enter key without inspecting the element for google Search.

↓ It will close the current window

To get the current windows By using Index :-

If we have 10 windows we get the required window using Index or pagestyle.

write the webElement for gmail

```
}
}
```

To get the window 2 :-

```
Set<String> pid2 = driver.getwindowHandles();
java.util.List<String> li = new ArrayList<>();
li.addAll(pid2);
String string = li.get(1);
driver.switchTo().window(String);
```
window 1 li.get(0);
window 2 li.get(1);

To Get the Gmail window & open signin :-
(using pagestyle)

```
for(String y : pid2){
    driver.switchTo().window(y);
    String title = driver.getTitle();
    if(title.contains("Gmail")){
        driver.switchTo().window(y);
        driver.findElement(By.xpath("(//a[contains(text(),'Sign in')])[2]")).click();
    }
}
```

target_blank — Opens new tab.

Frames :- demo.automationtesting.in/Frames.html
or pdf or doc

One html is embadded with in another html is called frames.

We should switch to the unframed window than only we can perform the actions. we can achieve it using following three.

Name
Index (or)
WebElement
} method overloading

```
WebElement iframe = driver.findElement(By.id("singleframe"));
driver.switchTo().frame(iframe);
driver.findElement(By.xpath("//input[@type='text']")).
                                    sendKeys("Hello");
}
```

Lab Table :- https://www.toolsqa.com/automation-practice-table/

```
<table>
    <thead>
        <tr>
            <th> </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td> </td>
        </tr>
```

To print All the data in td :-

```
WebElement table = driver.findElement(By.tagName("table"));   // to find single element
List<WebElement> tr = table.findElements(By.tagName("tr"));
for (WebElement x : tr) {
    List<WebElement> td = x.findElements(By.tagName("td"));   // to find multiple elements
    for (WebElement y : td) {
        String text = y.getText();
        S.o.pln(text);
    }
}
```

o/p :- It will print all the data inside td.

To print the particular data in td particular row :-

```
WebElement table = driver.findElement(By.tagName("table"));
List<WebElement> tr = table.findElements(By.tagName("tr"));
for (WebElement x : tr) {
    List<WebElement> th = x.findElements(By.tagName("th"));
    for (WebElement y : th) {
        String text = y.getText();
        if (text.contains("Taipei 101")) {
            List<WebElement> td = x.findElements(By.tagName("td"));
            for (WebElement z : td) {
                S.o.pln(z.getText());
            }
```

To check whether particular heading is present or not:

```
String City = td.get(1).getText();
if(City.contains("Taipei")){
    S.o.pln("pass");
```

To print all the td in the table :-

```
List<WebElement> td = driver.findElements(By.tagName("td"));
for(WebElement x : td){
    S.o.pln(x.getText());
```

Dynamic Webtable :- If the rows & columns changing dynamically then we can print using row index value or column index value.

To print row wise data :

```
List<WebElement> data = driver.findElements(By.xpath("//table//tr["+3+"]//td"));
for(WebElement x : data){
    S.o.pln(x.getText());
```

To print Column wise data :-

```
List<WebElement> data = driver.findElements(By.xpath("//table//tr//td["+3+"]"));
for(WebElement x : data){
    S.o.pln(x.getText());
```


TestNG :-

1. Annotation
2. priority
3. Invocation ~~comment~~ count.
4. ~~Enregrity~~ Ignore Test case
5. parameters
6. parallel Execution
7. Depends on methods.
8. Groups
9. Re execute the failed testcase.
10. Re execute the failure test case automatically.
11. Data provider.

Installation :-

1. Go to eclipse → Help → Eclipse Market place → Search for TestNG → Install Now → copy the path & Search in google → click → copy the url and Help → Install new software → Hosts → Maven paste the path → Next → finish
2. New project → New Maven project → Next → Next

Grp Id : com.TestNG , Artifact id → TestNG practice.

3. Google → testNG Maven dependency → 6.14.3 → copy maven dependency → Eclipse → pom → paste in dependencies → paste → copy Selenium java → paste inside pom under TestNG practice (dependency).

1. Annotation :-

Suite – collection of Test (Before tests)

Test – collection of class (run before each class)

Class (run before each Method)

Method (run before each Test)

Test (Actions to be performed)

Method

Class

Tests

Suite.

```
package

public class AppTest {
  @BeforeSuite
    public void beforeSuite(){
      s.o.pln("BeforeSuite");
    }
  @Before Test
      public void beforeTest(){
        s.o.pln("BeforeTest");
      }
  @Before Class
      public void beforeClass(){
        s.o.pln("BeforeClass");
      }
  @BeforeMethod
      public void beforeMethod(){
        s.o.pln("BeforeMethod");
```

```
  @Test
      public void test1(){ s.p.ln("Test1"); }
  @Test
      public void test2(){
          s.o.pln("Test2"); }
  @AfterMethod
      public void afterMethod(){ s.o.pln("AfterMethod"); }
  @AfterClass
      public void afterClass(){ s.o.pln("AfterClass"); }
  @AfterTest
      public void afterTest(){ s.o.pln("AfterTest"); }
  @AfterSuite
      public void afterSuite(){
        s.o.pln("AfterSuite");
      }
```

* Remove the junit packages from package. Then only it will import the package for Test annotation

Right click on .java → TestNG → convert to → It will open one xml → finish → testng.xml → re → run as testng.

O/P :-

Order of Annotation.

BeforeSuite
BeforeTest
BeforeClass
BeforeMethod
Test1
AfterMethod
BeforeMethod
Test2
AfterMethod
AfterClass
AfterTest
AfterSuite.

Before – pre condition

After → post condition

Without "@Test", we cannot run TestNG

2. priority => It will run in alphabetic order. To overcome this we can set priority.

priority => -ve to +ve.

If there is no priority it will take "0" by default

```
package
    class AppTest {
@Test (priority = -3)
public void Homepage() {
        S.o.pln("H.P");
}
@ Test (priority = -1)
    public void LoginPage() {
            S.o.pln("L.P");
@ Test (priority = 0)
    public void SearchProduct() {

    }
@Test   public void SelectProduct
    {     S.o.pln("selectproduct");
      }
```

3. ~~Invo~~ Invocation Count :-

To run the test case multiple times.

Ignore => It will ignore to run the particular test case.

```
package
    class
@ Test (invocationCount = 10)        => 10
    public void test1() {
        S.o.pln("Test1");
    }
    @ Test @ Ignore
        public void test2() {        => It will be ignored
            S.o.pln("T2");
        }
@
Test
    public void test3() {            => 1
        S.o.pln("T3");
    }
@ Test (enabled = false)             => Another method to ignore the test case is "enabled = false"
        public void test4() {
            S.o.pln("T4");
        }
```

O/P :-

Total count is (11)

Test1
Test1
Test1
Test1
Test1
Test1
Test1
Test1
Test1
Test1
Test 3

10

By default it will be true => True means it will be execute.

4. Ignore Test case :-

To ignore 3 types are available :-

1. @Test @Ignore
2. @Test (enabled = false)
3. In xml add method exclude.

3rd type to ignore a test case :-

Go to testng.xml

After class => add

```
<methods>
<exclude name="test 2"></exclude>
</methods>
</class>
```

If you have Test1, Test2, & Test3 it will ignore test 2 & print only Test1 & Test3.

If you want to run only one test case

```
<methods>
<include name="test 1"></include>
</methods>
```

A: 5. parameterisation :-

```
Class AppTest {
@Test
@parameters ({"username", "password"})
public void credentials (String un, String ps) {
    S.o.pln (un + ps);
}
}
```

testng.xml

```
<Suite name="Suite">
<parameter name="username" value="nagarjuna">
</parameter>
<parameter name="password" value="123xyz">
</parameter>
<test thread-count="5" name="Test">
<classes>
<class>
```

O/p :- nagarjune 123xyz

Optional :-

In testng if is username do mismatch occurs.

```
@Test
@parameters ({"userName", "password"})
Public void credentials (@optional ("Sudha")
                         String un, String ps)
    S.o.pln (un+ps);
```

If any mismatch occurs then it will take the optional value given.

O/p :- Sudha 123xyz

6. Parallel Execution :- Groups :-

We can execute more than one method using "groups" with same group name. Also we can use two group name in single group.

```
class
@Test (groups = "prr")
  public void vr() {
      s.o.pln ("vr");
}
@Test (groups = "prr")
  public void skywalk();
      s.o.pln ("skywalk");
}
@Test (groups = "sathyam")
   pub void sathyam() {
      s.o.p ("sathyam");
}
@Test (groups = {"inox", "prr"})
   pub void inox() {
      s.o.pln ("inox");
}
@Test (groups = "sathyam")
      pub void foram() {
         s.o.pln ("foram");
}
```

testng.xml

```
Suite
 <groups>
  <run>
   <include name = "prr"> </include>
  </run>
  </groups>
  <test thread
```

o/p :- vr
skywalk
inox

~~Groups~~

To run more than one group :-

```
<suite name = "Suite">
   <test thread_count = "5" name = "Test">
   <groups>
      <define name = "two">
         <include name = "prr"> </include>
         <include name = "Sathyam"> </include>
      </define>
      <run>
         <include name = "two"> </include>
      </run>
      </groups>
      <classes>
```

o/p :- vr
skywalk
inox
Sathyam
foram

7. Depends on Methods :-

It will run depends on method only.

```
@ Test
public void launchBrowser() {
    S.o.p ("Launch Browser");
}
@ Test ( dependsOnMethods = "launchBrowser")
public void signIn(){
    S.o.p ("SignIn");
}
@ Test ( dependsOnMethods = "signIn")
public void closeBrowser(){
    S.o.p ("Signout");
}
```

o/p will be same.

O/P:- Launch Browser
Signin
Signout

8. parallel Execution :-

thread-count = "5" => At a time we can run 5 windows.

If the count is 2 but we define 3 methods means 1st & 2nd will open 2 browsers only and it will open the 3rd browser once any one of the browser completes.

```
@ Test
public void amazon(){
    System.setproperty ("webdriver. ", "E:\\Selenium\\Chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get ("https://www.amazon.in");
}
```

like this write 3 Test

testng.xml

```
<Suite name = "suite" parallel = "methods" >
```

If we want to execute 2 classes then give parallel = "classes"

Data provider :-

```
@ Dataprovider (name = "credentials")
public Object[][] credn() {
    return new Object[][] {{"sudha", "xyz123"},
                           {"Mathi", "abc124"}};
}
@ Test (dataprovider = "credentials")
public void facebook( String un, String ps){
    Sys. setproperty
    Webdriver driver = new ChromeDriver();
    driver.get("https://www.face
    driver.findElement(By.id("email")).SendKeys(un);
    driver.findElement(By.id("pass")).SendKeys(ps);
}
```

o/p :-

1st it will open one browser & Enter the given details.

Once it completes it will open another browser for another set of values.

Assert :- It will show how many methods are executed.

```
Class { @Test
        Public void test1() {
  Assert.assertEqual(actual, expected);
 }
```

index.html → rc → open with browser.

Assert.fail() ⇒ It will fail that particular test case.

Retry Analyzer :-

To execute the failure test case :-

Create a new class RetryAnalyzer.

interface ⇒ Should implement the interface & Override all the abstract methods in the implements class.

RetryAnalyzer.java

```
public class RetryAnalyzer implements IRetryAnalyzer{
  int minCount=0;
  int maxCount=5;
 public boolean retry(ITestResult Result){
  if (minCount<maxCount){
   minCount++;
   return true;
  }
   return false;
 }
}
```

AppTest.java

```
class AppTest{
 @Test(retryAnalyzer=RetryAnalyzer.class)
  public void test1(){
      S.o.pln("pass");
  }
 @Test(retryAnalyzer=RetryAnalyzer.class)
  public void test2(){
     Assert.fail();
     S.o.pln("pass");
  }
}
```

If there is too test cases.

Create an class called IAnnotationTransformer. Iannot

Iannot.java

```
Public class Iannot implements IAnnotationTransformer{

public void transform(ITestAnnotation annotation, Class
                testClass, IRetryAnalyzer ~~retry~~);

IRetryAnalyzer retryAnalyzer = annotation.getRetryAnalyzer
                ();
  if (retryAnalyzer == null){
    annotation.setRetryAnalyzer(RetryAnalyzer.class);
  }
}
```

testng.xml

We need to add listener in testng.xml

```
<Suite name="Suite">
 <listeners>
 <listener class-name="org.testng.maven.TestNG-
                practice.Iannot"></listener>
 </listeners>

   <test
      <classes>
         <class
```

06/07/2019

We cannot directly inspect the alert window. so we have to switch to the alert window first and then have to perform the action.

Alerts :-

Three types.

1. Simple Alert => Click - OK
2. Confirm Alert => click - OK, Cancel
3. Prompt Alert => click - Enter, OK, Cancel.

OK - Accept() ; Cancel - dismiss()

```
public class Alert {
 public static void chromeBrowserLaunch() Throws
                InterruptedException {
 System.setprop
 WebDriver driver = new ChromeDriver();
 driver.get("https://www.seleniumeasy.com/test/
                javascript-alert-box-demo.html");
```

Simple Alert :

```
 WebElement alertButton = driver.findElement(By.xpath("
                //button[@onclick='myAlertFunction()']"));
 alertButton.click();
 Thread.sleep(2000);
 Alert alert = driver.switchTo().alert();
 alert.accept();
 driver.switchTo().defaultContent();
```

Confirm Alert :

```
 WebElement confirmButton = driver.findElement(By.xpath(
                "//button[@onclick='myConfirmFunction()']"));
 confirmButton.click();
 Thread.sleep(2000);
 Alert alert2 = driver.switchTo().alert();
 alert2.dismiss();
 driver.switchTo().defaultContent();
```

Prompt Alert :

```
WebElement promptButton = driver.findElement(By.xpath("
        \\button[@onclick='myPromptFunction()']"));
promptButton.click();
Thread.sleep(2000);
Alert alert3 = driver.switchTo().alert();
alert3.sendKeys("Hello");
alert3.accept();
driver.switchTo().defaultContent();
}
public static void main(String[] args) Throws InterruptedException {
    ChromeBrowserLaunch();
}
}
```

Base class :-

Create Maven project . Unda Maven project create packages and Baseclass (class).

Ne have to add try catch block for each & every method to

```
public class Baseclass {
  public static WebDriver driver;
public static WebDriver (String getBrowser(String browserName)
                                throws Exception {
  try {
    if (browserName.equalsIgnoreCase("chrome")) {
    Sys.setprop("name", "path");
    driver = new ChromeDriver();
  } else if (browserName.equalsIgnoreCase("firefox")){
      sys.setprop
      driver = new FirefoxDriver();
  } else if (browserName.equalsIgnorecase
```

Methods :-

1. Browser launch
2. geturl
3. elementIsDisplayed
4. elementIsEnables
5. elementIsSelected
6. clickon the Element
7. element SendKeys
8. WaitforElementVisibility (Call this method inside all the other methods)
9. mouseOverTo the Element
10. dragand drop
11. dropdown (Based on index, value, visible Text)
12. getText

13. getAttribute
14. ScrollUp/ScrollDown ScrollUsingElement
15. ScrollUsingCo-ordinates.
16. TakeScreenShot

Selenium Grid :- Hub-node Architecture.
1-Hub multiple node

2. Disable firewall
firewall & network protection
Disable all firewall in all network.

3. cmd → ipconfig
IPv4 Address → our address

open another cmd
ping 192.168.43.142 (anyone's Ip id)
it will show "Reply from". (connecting with other lap)

4. open the local folder (for hub) C4-4)
1. Go to local folder → type cmd → cmd → then type the below
2. java -jar selenium-server-standalone-3.141.59.jar -role hub
3. Enter
default 4444.
4. Google => localhost:4444 - Enter.
connected.

(for Node)
< dac\selegrid>java -Dwebdriver.chrome.driver="copy the chrome path from the local folder you created".
(prev tab) created ".
-jar selenium-server-standalone-3.141.59.jar
hub id
role node hub http://192.168.43.143:4444
(should comes to client)
/grid/register.

Error.
Node is registered to hub and ready to use.

Hub: Go to baseclass (for hub)
under getBrowser

else if (browserName.equalsIgnoreCase("gridChrome")){
DesiredCapabilities Chrome = DesiredCapabilities.chrome();
Chrome.setPlatform(Platform.WIN10);
webdriver driver = new RemoteWebDriver(new url(http://192.168.43.4444/wd/hub"), chrome);}

copy gridChrome => configuration properties → paste.

1. To run multiple test case across multiple machines having different os & different browser.
2. Which takes time.
open a class SeleniumGrid {
→ DesiredCapabilities to webdriver.
driver.get("http://www.google.com");

07/07/2019

Git Hub :-

Git add
Git commit
Git push

To manage and share the code across all persons.

It is a Source code management tool and It will keep track of the versions so we can go to the previous developed codings
Distributed Version control System.

Goto Chrome → Git scm download → Click on download for windows → Run exe → Next → Next → Install → launch Git → Finish → It will click OK for browser → Click on 1st link in the browser → Sing up → verify using gmail → Start a project → Rep Name : Adactin-Automation → public → Create Repository

→ Google - Github.com => Cloud or Remote.

open pom in local drive (C: prog files / users / you. - selenium)

Right click → click on "Git Bash here" → It will open cmd :

Type the below commands

=> git config --global user.name "Sudhamathi14081990"
Enter

=> git config --global user.email "r.n.sudhamathi@gmail.com"
Enter

=> git init
Enter

=> git remote add origin https://github.com/Sudhamathi14081990/Adactin-Automation.git
Enter

=> git config --list
Enter

=> git status
Enter

=> git add pom.xml
Enter

=> git status
Enter

=> git add .

=> git status

=> git commit -m "message"

=> Create one more file in previous folder

14/07/19

getText()
getAttribute() => only valid for "Input" Tagname.
getTagName()
getOptions() => To print all the options under drop down
isDisplayed()
isEnabled()
isSelected()
Implicitwait()

Explicit wait
Actions
Select } parameterized constructor.

Implicit wait
Explicit wait } - Method overloading.

Sing in → Give Repository name → Automation Practice.
→ Description (optional) → public → ~~click~~ uncheck "Initialize this repo / if you're importing repository with a README".
proj desc, setup instructions, how to follow.

→ Now it will create new repository.

Cloud setup is done.
We have local code. Now need to setup the connection.

20/7/2019

Maven :- (Initialization in previous page).

1. Maven is a powerful project management tool that is based on pom (project Object Model).
2. It is used for projects build, dependency and documentation.
3. Maven will manage dependencies.
4. It creates default folder Structure.

pom (page Object Model) :-

1. To reuse the webelement and maintain the web element.
2. If any web element name is changed we can change it in pom it will change all the elements.

4 Tags in POM :-

1. FindBy - particular element.
2. FindBys - can define 2 element & both should match.
3. FindAll - can define all the elements & any one should match.
4. CacheLookup

Step 1 : (in pom) for each page.
Initialize the web driver
Step 2 : create constructor and assign the (driver to driver)
Step 3 : Initiate elements using page factory.
Step 4 : Define web elements @FindBy
Step 5 : Source -> Getters & Setters.

Maven

Src/main/java => Base class & pom
Src/test/java => testscripts.
Src/test/resource => Drivers.
Src -> pom.xml => Selenium java (3.141.59)
commons-io (2.6)
Junit (4.12)
} dependencies

Junit :-

It is a unit testing framework.
Add junit dependencies in pom.xml (remove Scope - test)

Annotations :-

@BeforeClass - It will execute before all @Before method (should use static)
@AfterClass - It will execute after all @After Method (should use static)
@Test - Test code
@Before - It will execute before each test.
@After - It will execute after each test.

If you have multiple test @Before method will run "Before each test".

Selenium Automation tool

Class {

main(){

}

}

(1)

methods

(2)

Pom.

class HomePage{

@FindBy

Eg.

}

class Register(){

}

Class A extends baseclass{

main()

Testuri.(h2...&)

}

Framework.

BDD →

Behaviour Driven Dev...

Feature

Feature:

Scenario

Given. When. Then. and

Scen...

Base class

Use logic or ?

getter Browser Conn(...)

plain English

(Gherkin)

Selenium - java Action.

Runner.

@ RunWith Cucumber. class

@ Cucumber Options { Features.

glue : S.D.

plugin.

}

Be Driven Develop.

29/7/19

Adv :- Test data & Test Script will be in same feature file.

language - Gherkin

Plugin - Natural

Cucumber :-

Natural

Install Gherkin Plugin → Help → Market place → (Search for Cucumber / gherkins) - Natural 0.7.6 (install) → confirm → Accept - finish

Add the below dependencies in Pom.xml

1. Selenium java (3.141.59)
2. junit - 4.12
3. Cucumber java - 1.2.5 (info.cukes - older version bcoz new version will not support)
4. Cucumber junit - 1.2.5
5. Cucumber core - 1.2.5
6. common io - 2.6

Steps to create cucumber project :-

Create new maven project → Under src/test folder create a new file as "Automation Practice.feature" (Feature file contains test cases in plain english).

1. Feature - Multiple Scenarios
2. Scenario - Purpose of particular test case
3. Given - precondition (already given)
4. When - Actions to be performed
5. And - following action
6. Then - for validation.
7. But - for Negative projection
8. * ⇒ Instead of Given, And, when, Then

10/8/19

Excel Read :-

```
package com.excel.readwrite;

public class ExcelRead {
public static void main (String[] args) throws IoException {
File f = new File ("D:\\Excel.xlsx");
FileInputStream fin = new FileInputStream (f);
// .xlsx => XSSFWorkbook (or) .xls => HSSFWorkbook
Workbook wb = new XSSFWorkbook (fin);
Sheet Sheet = wb.getSheet ("Sheet1");
int PhysicalNumberOfRows = Sheet.getPhysicalNumberOfRows();
Sys.out.println (PhysicalNumberOfRows);
for (int i=0; i< PhysicalNumberOfRow ; i++) {
  Row row = Sheet.getRow(i);
for (int j=0; j<row.getPhysicalNumberOfCells(); j++){
  Cell cell = row.getCell (j);
  CellType cellType = cell.getCellType ();
  String cellValue = null;
  if (cellType.equals (CellType.STRING)){
    cellValue = cell.getStringCellValue();
} else if (cellType.equals (CellType.NUMERIC)){
  double numericCellValue = cell.getNumericCellValue ();
    long l = (long) numericCellValue; => Type casting
    cellValue = String.valueOf (l);
Sys.out.println (cellValue); } }
```

It will return 10.00 in normal practice

17/8/2019

1. Easy to debug
2. We can reuse the methods (Steps) Auto suggestable
3. Developing the actions into code.

Cucumber :- (Behaviour Driven Development)
(converting the application behaviour into scripts)

1. Install "Natural 0.7.6 from Eclipse Marketplace

2. Three folders.
   a. Src/main/java
      1. com.cucmber.baseclass
      2. com.cucumber.pom
   b. Src/main/test/java
      1. com.cucumber.StepDefinition
      2. com.cucumber.TestRunner
      3. Create one feature file. (com.cucumber.featurefile)
   (Green colorion) (extension => .feature)
   c. Src/test/resources.
      1. com.cucumber.drivers.
         para driva.

① TestRunner.java.
(Annotation) → we need to run with junit
@RunWith (Cucumber.class)
class.

@CucumberOptions (
features = "feature file path from src" ,
glue = "stepdefinition package path"
)

project => Framework - Singleton design patrern.
extend reports.
Real time methods
property file
Selenium Grid
GIT
Jenkins
Agile
Jira
Software Testing
Mobile Testing using APM
API Testing
Interview preparation.

project - Cucumber

frameworks => POM, Junit, TestNG, Datadriven

Step parameter :-

(for particular id).

Scenario Out :
Given User launch the amazon Apple.
Add User click on the Sign in button in the Sign in module.
And User Enter email id "Sudha@gmail.com" in sign in page.
And User Enter password "abc123" in the sign in page
And User click on the login button in the login page.
Then User verify the UserName "Sudha" displayed in the header.

Scenario Outline : (Same test case for multiple data)

Background : pre requisite for each & every Scenario

Singleton (Design pattern) (method) :-

1. com.cucumber.util

a. pageObjectManager.java => create class variable use (or) getters -> create constructor (for class)

Creating only one instance for each class and reuse that instance wherever required.

Tags :- (before Scenario) To run particular Scenario give a name to each Scenario

Eg : @Tag1

Mention this tag name in Test Runner to run the particular Scenario.

Tags before Feature :-

If we have two feature files & we want to run one among them, we need to name the feature file.

Eg : @Sanity
Feature : Amazon Automation Sanity.

Add this tag name in Test Runner

Eg : tags = "@Sanity". It will run that particular feature file.

To run two Scenarios : (It is like 'or' cond)

tags = "@TC1, @TC2" => It will run only TC1 & TC2 Scenarios.

@TC1 @TC2 @TC3 } we can give more than one tag also.
Scenario :

If we want to run more than two Scenarios :-

We can give as common tag for the required Scenarios & we can use that common tag name in Test Runner class.

Eg :-

tags = "@DEMO" => It will run that particular Scenarios with that common tag.

If we don't want to run particular Scenario :

Eg :- @TC2 @demo @Ignore

Scenario outline :

we need to give @Ignore before that Scenario & mention that in Test Runner.

tags = {"@DEMO", "~@Ignore"}

AND Condition :-

→ Separate " " for two cond.

tags = { "@DEMO", "@TC1", "~@Ignore" }.

Both should satisfy.

OR condition :-

tags = { "@DEMO", @TC, " }

single double quotes.

To Generate reports :-

1. Create a package under src/test/resource as "com.cucumber.reports".

2. In Test Runner.java add the below paths in @CucumberOptions to generate the html, xml, json reports.

@CucumberOptions (plugin = { "pretty", "json : src\\test\\resources\\com\\cucumber\\reports\\report.json", "junit : src\\test\\resources\\com\\cucumber\\reports\\report.xml", "html : src\\test\\resources\\com\\cucumber\\reports" + "*" },

features = "package name of feature file", glue = "package name of Step definition" }).

(→ to generate the report in proper format. → path of report package. → file name → json report → xml report → file name → html report → file name not needed for html report)

3. After adding these path run the program and refresh the reports package. Now we can see the reports generated.

html → open with → web browser

json, xml → open with → Generic Text Editor.

property file :-

we need to run the url from developer, testing and live. So on that time each url will get changed. To overcome that issue we are using property file.

com.selenium.properties (package)

Configuration.properties (file)

Under ~~Config~~ File :

BrowserName = Chrome.

Url = http://automationpractice.com/index.php?

Goto util → Create a class (ConfigReader)

Create one constructor

Create a variable for "File"

```
public class ConfigReader {
    public Properties pr = new Properties();
    public ConfigReader() {
File propfile = new File("path of file");
FileInputStream fis = new FileInputStream(fis);
        pr.load(fis);
```

(pre defined class)

Surrond with try / catch block

Create two methods for Browser Name, Url.

in

Create one class and util.

=> public class FileDataManager {

Create class level variable.

=> public ConfigReader cr;

1. Configuration.properties. (file)

Browser Name = Chrome

Url = url you want to load

2. ConfigReader (Class) Also create cons to read prop file

Create obj for prop file & Create two methods for BrowserName & url

3. FileDataManager (Class)

Create obj for FileDataManager class

Create obj for ConfigReader Class (to use like Singleton method)

=> getters

=> return

Declare FileDataManager as private Constructor (No one should create Obj for this class so declaring it as private)

4/9/15

extentreports.com/docs/versions/4/java/v3-html-reporter.html

Extent Report :-

Under com.cucumber.properties

-> create one file.

extent-config.xml

copy the entire dependencies from above url & pasre it in "extent-config.xml" file.

We need to add two dependencies.

=> Google -> Extent report maven dependency

-> ~~Central repository~~ search -> Cucumber-extents report 3.0.2

extentreports 3.0.6

TestRunner -> (plugin = {"com.cucumber.listener.(Extent

maven dependencies

Cucumber Extent reports

Cucumber formatter): pasre the path of Extent-reports\extentreport.html

com.cucumber.listener

ExtentCucumberFormatter.class

remaining).

Add one package for extent report.

In TestRunner @AfterClass add the below line.

```
Reporter.loadXMLConfig(new File(System.getProperty
("user.dir") + " path of extent-config.xml"));
```

After running the code refresh the extent report package. open the html file with web Browser. It will show the no.of features, scenarios, passed Testcases (green), failed Tcs, undefined Test cases.

Under Extent report package create one folder package for Screenshot.

To take the Screenshot of the failed testcase:

Add the below line in Hooks.java class. Hooks will run before & after each method.

We should return the dest file in the Screenshot method in Baseclass. Then only we can take Screen shot.

```
Scenario.getStatus();
if (Scenario.isFailed()) {
File screenshot = screenshot (Scenario.getName());
Reporter.addScreenCaptureFromPath(screenshot.
                    getAbsolutePath());
```

(name of the Scenario)

If we want to take Screenshot after any step in method then add the below line.

```
File screenshot= screenshot ("user_click_on_the_
                    product");
Reporter.addScreenCaptureFromPath (screenshot.
                    getAbsolutePath());
```

Highlight :-

To Highlight the element which is used in Assert.

```
public static void highlightWebelement() {
JavaScriptExecutor js = (JavaScriptExecutor) driver;
js.executeScript("arguments[0].setAttribute('style',
    'background
```

Add the below line in stepdefinition wherever want :- (Assert)

```
highlightWebelement(pm.getOp().getTopSeller());
```

(name of the element)

Screenshot for particular passed Testcase :-

To Take Screenshot for passed Method :-

```
File screenshot = screenshot ("method name");
Reporter.addScreenCaptureFromPath (screenshot.getAbsolute
                    path());
```

Assertion :- Used to compare the actual result of an application with the expected result.

i) Hard Assertion : Does not continue with execution until the assertion condition is met.

ii) Soft Assertion : Continues with test execution even if the assertion condition is not met.

(Need to know about SDLC & Agile)

Jenkins :- (Refer Notes in laptop)

Google → Jenkins → 1st link → click on download → Click on "Generic Java package (.war)" to download.

Create New folder → paste war file → rc → get Bash here, type ↓

Java -jar Jenkins.war (Don't close otherwise Jenkins will not run)

⇒ Jenkins is fully up & running

Google → localhost:8080 → Enter → copy the link → run → open with notepad → OK → copy password & paste in Chrome → Continue → Install Suggested Jenkins

→ It will take 5 to 10 mins to install → Give all details as admin & E-mail as admin@gmail.com → Save & continue → finish → & Click on Start Jenkins.

Manage Jenkins → Global Tool configuration → Add JDK → JAVA_HOME → Environment variable, Maven_Home → paste Maven path.

Enter

Admin Automation → Freestyle project → OK

General : Description

Source code : Give git url

Branch : master

Build Environment : To delete workspace (previous code)

Build : Maven, Version → Maven_HOME → Goals clean test

post build Actions : Add post build → Save

Build now → open console.

Back to Dashboard → Manage Jenkins → manage plugin → Go to Available tab → Search for cucumber → Install without restart.

Goto dashboard → Click on project → Click on configure → post build Actions → Cucumber reports → Advanced → JSON path → paste json path.

→ Build Now, push the updated Testrunner using git → git add . → git commit -m " " → git push → Build Now.

Go to cucumber reports → Click on feature name → It will show the report (we can see the previous reports as well).

Build Trigger → Build (Smoke test will run auto) after project is build → Deployment project → Run whenever deployment happens → Save. → New Deployment project

2nd case:

Configure → Build Triggers → Build periodically (Check off) → Schedule → ? → Minute Hour Dom Month Dow

H/15 * * * *

It will run every 15 mins.

H (0-29)/10 * * * *

Every 10 mins in the first half of every hour. (3 times)

Once every two hours at 45 mins past the hour starting at 9.45 am & finishing at 3.45 pm every weekday.

45 9-16/2 * * 1-5

H/15 * * * * (To run in next 5 mins).

Jenkins:

Whenever Deployment is going to happen Smoke test will be triggered automatically (based on when Build after project build) & it will do regression testing based on client requirement based on Build periodically.

Developer → Build the appln → package → unit test → war → code coverage → deploy in dev → test execution → report → share.

dev → 5 step for Test Server → test execute → report → share.

prod → test.

Continous Integration → Continuous Deployment → Continuous Test

CI CD CT

1. Data driven interface.

It is used to run one automation test script with huge amount of testing data.

Where code stays the same & data input keeps changing. useful in regression kind of scenarios.

2. Why webdriver as an interface?

Since, selenium developers does not know logic & logic will not be same for all the browsers, they created WebDriver as an interface It is a type in java which has only abstract method i.e., methods without body / logic.

If no driver location found ⇒ IllegalStateException

1 byte = 8 bit.

Data Types. < Primitive / Non-primitive.

Wrapper class

DS.	Primitive.	memory size (byte)	default value	Wrapper class	Non Primitive
1.	byte	1	0	Byte	
2.	short	2	0	Short	
3.	int	4	0	Integer (ex.)	String
4.	float				Array
4.	long	8	0	Long	
5.	float	4	0.0f	Float	etc. = OS,
6.	double	8	0	Double	
7.	char	2	'\u0000'	Character (ex.)	
8.	boolean T/F	1	(ex.) False.	Boolean	

numbers.

decimal

1. Project
2. Interface
3. Array & Array List
4. String builder & Buffer
5. String program
6. Dependencies of TestNG
7. Excel data Driven code
8. Rate yourself in Java (4)
9. Framework Explanation (BDD - Cucumber)
10. Roles & Responsibilities
11. HashMap.
12. Setter & getter Class.
13. BDD - framework
14. Feature file creation
15. Step definition.
16. Agile
17. Hashmap, when will you use.
18. How to get a value, available at particular Hashmap.
19. Which code repository you are using.

---

1. Declaration & Definition in Java
2. How to achieve runtime polymorphism? → method overriding
3. How to take Screenshot
4.

Buffer - Sync.

byte

Range:

-2^{n-1} to $(2^{n-1}-1)$

-2^{8-1} to $(2^{8-1}-1)$

$\Rightarrow -2^7$ to 2^7-1

$\Rightarrow -128$ to 127

Parthiban
Vinoth

Megha
Dilip
Pradeep
9566277878

Pending Topics :-

Drag & Drop
Right click & Double click, context click
Robot class
Screenshot
Alert
windows Handling
web table
Javascript
file
Base Class

Adding Branch from Master

git checkout -b "SudhaBranch"
Enter
git add .
Enter
git commit -m "message"
Enter
git push origin SudhaBranch
Enter
Username & password
ok

git remote remove origin
git remote add origin https://github.com/arulselvi123/Adactinautomation
git add .
git commit -m "message"
git push origin SudhaBranch
username & password → ok

```
for (i=1 ; i<=3 ; i++){
    for (j=1 ; j<=3 ; j++){
        SOP (j)
    }
    SOPln();
}
```

o/p

123
123
123

Trim :- It will remove the space infront of & after the last doctor.

```
s.trim()
```

Contains :- To verify whether any particular char or word is present in a word.

```
String s = "Hello ABC";
s.contain ( "ABC");
```

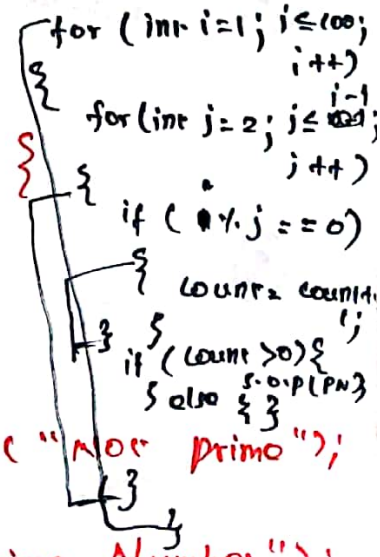
if any Id is changing dynamically we can use the words alone in contains.

Concatination :- we can use + also "Hello + world";

```
String x = "Hello";
String y = "World";
String concat = x.concat(y);
```

Prime Number :-

```
public class PrimeNumbers() {
public static void main (String[] args) {
    Scanner sc = new Scanner (System.in);
    S.o.pln ("Enter a Number");
     n = sc.next2nr();
Int count = 0;
  for (int i=2 ; i≤ n-1 ; i++) {
     if ( n % i == 0) {
      count = count+1;
     }
}
     if ( count >0) { s.o.p ("Not Prime");
                    }
     else { s.op ("Prime Number");
}  }
```



Php Travels

1. Goto php Travels click on Sing up button. Enter data & Sing up. (see out) &
2. Verify the displayed user name.
2. Search for hotels, check in date, out, adult, child → Search

Orange HRM

1. Login the appln & get data from db.
2. Recruit a candidate : select all the drop downs

Make my Trip

(1) login with username & password
Database - JDBC -

(2) Search flight for 3 scenarios.
using Example.

(3) my account add address
↓
Excel

(4) Click on hide now which has cursor pointer value
- Custome

(5) Verify same flight selected
↓
Collection

To Automate windows based appln :- (AutoIt)

Download AutoIt
Download AutoIt editor.
Prog file - AutoIt → SciTE (opens one editor).
Run AutoIt3 AutoInfo → drag finder tool in file name (to upload file).

In SciTE → (title, text, control Id = class + Instance)
ControlFocus("Open", "", "Edit1")
ControlSetText("Open", "", "Edit1", "C:\Users\Sudha\Documents\.docx")
ControlClick("Open", "", "Button1") => drag finder tool & place in open button.
=> Save it as fileupload → rc → compile script → Run will

In eclipse :-

driver.findElement(By.id("file")).click();
Runtime.getRuntime().exec("D:\fileupload.exe");

```
title = driver.getTitle();
    S.o.ph(title);
Set<String> pid = driver.getWindowHandles();
for (WebElement x : pid)
{
    driver.switchTo().window(x);
if (x.equals("Gmail")
{
driver.switchTo().window(x);
}
else {
    S.o.pln("Title not found")
}
```

```
driver.switchTo().iframe(element for iframe)
driver.findElement(By.xpath(for un)).sendKeys("Sudha");
driver.findElement(By.xpath(for pwd)).sendKeys("xyz");
driver.findElement(By.xpath(for ok)).click;
driver.switchTo().defaultContent();
driver.findElement(By.xpath(for ok btn)).click.
```

```
table = driver.findElement(By.tagname("table");
List<String> th = driver.findElement(By.tagname("th");
for (WebElement x : th)
{
if (x.equals("column2") || x.equals("column3")
{
    S.o.p(x.getText());
}
```

```
title = driver.getTitle();
    S.o.ph(title);
    cid = driver.getWindowHandle();
using Robot class open 4 windows.
Set pid = driver.getWindowHandles();
for driver.switchTo().window(pid);
for (pid : pid) {
if (pid.contains("Gmail")
{ driver.switchTo().window(pid);
}
}
```

Eclipso → rc → Show in → Sys Explora → open seperate folder → rc → Git Bash Here.

1. Go to Git Bash Here
2. `git --version` ⇒ It will the git version.
3. `git config --global user.name "Sudhamathi 14081990"`
4. git config --global user.email "r.n.sudhamathi@gmail.com"
5. To Check the setup: It will show the username & user email.
   `git config --list`
6. To tell which repository we are going to use,
   ⇒ git init (To tell the git about the folder to initialize)
   ⇒ git remote add origin https://github.com/Sudhamathi1408l990/Adactin Automation.git
7. To file all the files & folders in cucumber.
   `ls`
8. git add pom.xml (to add seperate file)
9. git status (It will show one new file pom.xml)
10. git add . (To add all the file in one shot)
11. git status (It will show all new files).
12. To tell git why we are going to push particular code.
    `git commit -m "to add cucumber project".`
    (-m → message.)
13. To push the code to remote:
    git push -u origin master
    (-u → upstream branch for the first time; origin → name of github; master → code from master System.)

⇒ Go to Github & refresh it will show all the codes in repository.

⇒ -u is for the first time push only. If u use for next time it will get rejected. So for that, you need to pull the code from repository then only we can push again.

⇒ add one new file in the git bash folder.

⇒ git add .

⇒ git commit -m "Added some file in git bash folder".

⇒ git push (To push the updated folder

To pull the user data from Someone's repository:

1. Search the user id
2. Click on the link to download
3. Create new folder in local folder.
4. Go to clone or download & copy the url

   git clone paste the clone url

   It will pull all the data from that user & saved in the new folder created in your local folder.

If we not able to perform any operation through Selenium, we can use

1. Javascript
2. Robot Class

Last promotion => July 1st.

Variable pay => nothing

Fixed pay & Gross CTC — Same.

Variable payout pattern — Annual Income.

expected 30%. like.

Client location — Resume comp

2012 feb

2013 - 14
2014 - 15
15 - 16
16 - 17
17 - 18
18 - 19

} 6 years

4 yrs in automation
8 to 8.5

1 proj - 35 sprints
1.7
1.9

1st Sprint

=> poc (proof of concept) => By analysing the req & we can give some ideas about How to implement.