**SELENIUM**

**1**. **Multiple window handling:**

driverRefname.switchTo.getWindowsHandles();

**2. Positive case:**

**Using valid username and valid password:**

public class Alert{

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

Webdriver driver=new chromedriver;

driver.get(“https://www.facebook.com/”);

Webelement email=driver.findelement(By.id(“email”);

email.sendkeys(“javaselenium@gmail.com);

Webelement pass=driver.findelement(By.id(“pass”);

pass.sendkeys(“javaselenium”);

driver.findelement(By.id(“login”).click();

JavascriptExecutor obj=(JavascriptExecutor)drive;

obj.executeScript("alert('Welcome to facebook')");

drive.switchTo().alert().accept();

**Negative case:**

**Using valid username and Invalid password:**

public class Alert{

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

Webdriver driver=new chromedriver;

driver.get(“https://www.facebook.com/”);

Webelement email=driver.findelement(By.id(“email”);

email.sendkeys(“javaselenium@gmail.com);

Webelement pass=driver.findelement(By.id(“pass”);

pass.sendkeys(“java”);

driver.findelement(By.id(“login”).click();

JavascriptExecutor obj=(JavascriptExecutor)drive;

obj.executeScript("alert(‘incorrect password’)");

drive.switchTo().alert().accept();

**3**.**Code for CrossBrowser Testing:**

Public class crossbrowsertesting{

Webdriver driver;

@BeforeTest

@Parameters(“browser”)

public void setup(String browser)throws Exception{

if(browser.equalsIgnorecase(“chrome”))

{

System.setproperty(“path of chrome driver”);

driver=new Chromedriver;

}

elseif(browser.equalsIgnorecase(“firefox”))

{

System.setproperty(“path of gecko driver”);

driver=new Firefoxdriver;

}

elseif(browser.equalsIgnorecase(“edge”))

{

System.setproperty(“path of edge driver”);

driver=new Edgedriver;

}

else

{

}

**4**. **To handle frame:** driver. switchTo ().frame();

**By index :** driver. switchTo ().frame(int index);

**By id:** driver. switchTo ().frame(String id);

**By webelement:** driver.Switchto().frame(Webelement element);

**5.**public static void main(String[] args) throws AWTException {

WebDriver drive=new ChromeDriver();

drive.get("http://www.xyz.com/");

Actions a=new Actions(drive);

WebElement element = drive.findElement(By.*id*("menu"));

a.click(element).build().perform();

WebElement element2 = drive.findElement(By.*id*("submenu"));

a.click(element2).build().perform();

WebElement element3 = drive.findElement(By.*xpath*("//a[@href=’link’]"));

a.click(element3).build().perform();

WebElement element4 = drive.findElement(By.*id*("element"));

a.click(element4).build().perform();

**6**. **By using getAllSelectedoptions() method we can print selected options.**

Select refName = new Select();

refName.getAllSelectedoptions();

**8**. **Screenshot syntax:**

TakeScreenshot takescreenshot= (TakeScreenshot)driver;

File screenshotAs=takescreenshot.getscreenshotAs(OutputType.File);

File f=new File(“file with path”);

FileUtils.copyFile(screenshotAs,f);

**9. In Excel sheet:**

**LOGIN TESTCASE**

|  |  |
| --- | --- |
| **USERNAME** | **Java** |
| **PASSWORD** | **Selenium** |

1. **In src/main:**

* **Baseclass:**

**(Test data driven in base class)🡪data passing**

Public class baseclass{

static String *value*;

public static String data\_Pass(String s,int r,int c) throws IOException {

File f=new File("path.xlsx");

FileInputStream fi=new FileInputStream(f);

Workbook w=new XSSFWorkbook(fi);

Sheet sheetAt = w.getSheet(s);

Row row = sheetAt.getRow(r);

Cell cell = row.getCell(c);

*CellType* cellType = cell.getCellType();

if (cellType.equals(*CellType*.***STRING***)) {

System.***out***.println(cell.getStringCellValue());

*value*=cell.getStringCellValue();

}

else if (cellType.equals(*CellType*.***NUMERIC***)){

double numericCellValue =cell.getNumericCellValue();

int numeric=(int) numericCellValue;

System.***out***.println(numeric);

*value*=String.*valueOf*(numeric);

w.close();

}

return *value*;

}

}}

**// sendkeys**

public static void sendkeys(WebElement element, String send) {

element.sendKeys(send);

}

**// click**

public static void click(WebElement element) {

element.click();

}

* **Pom class:**

public class Pom\_Login {

public WebDriver driver;

*@FindBy*(id="user-name")

private WebElement user;

public WebElement getUser() {

return user;

}

*@FindBy*(id="password")

private WebElement password;

public WebElement getPassword() {

return password;

}

*@FindBy*(id="login-button")

private WebElement click;

public WebElement getClick() {

return click;

}

public Pom\_Login(WebDriver driver2) {

driver=driver2;

PageFactory.*initElements*(driver, this);

}

}

1. **In src/test class:**

public class Login extends Baseclass {

public static WebDriver *driver*=*launch*();

Pom\_Login obj=new Pom\_Login(*driver*);

*@BeforeMethod*

*@Test*(priority = -1)

private void URLlaunch() throws IOException {

*url*("https://www.facebook.com/");

}

*@Test*(priority=0)

private void Login() throws IOException {

String data1 = *data\_Pass*("login testcase",0,0);

*sendkeys*(obj.getUser(), data1);

String data2 = *data\_Pass*("login testcase",1,1);

*sendkeys*(obj.getPassword(), data2);

*click*(obj.getClick());

}

**10.Test scenario for Credit Cards:**

* Verify the credit card is having 16-digit number.
* Verify the expiration date should be displayed on the credit card.
* Verify the card holders name should be in a correct format with correct spelling.
* Verify the bank name should be displayed on the credit card.
* Verify the bank logo should be displayed on the credit card.
* Verify the cvv number should be displayed at the back side of the credit card.
* Verify the color of the credit card.
* Verify the user should be able to do the transaction using credit card.

**11. New Window:**

drive.switchTo().newWindow(*WindowType*.***WINDOW***);

**12.** Using Actions class we can switch from one tab to another tab and by using getwindowshandles.

**15.** By using findelements, we can find more than one web element in the list.

\*\*\*\*\*\*\*\*

**JAVA Q/A**

**2.** First we must identify the length of the string and to traverse till its length.

Then, extracting each character and add those character in front of existing string.

**3**. **Detect duplicate characters in a string:**

class Main {

public static void main(String[] args) {

String str = "Hello World";

for(int i = 0; i < str.length(); i++) {

for(int j = i+1; j < str.length(); j++) {

if(str.charAt(i) == str.charAt(j)) {

System.out.println("Duplicate Character: " + str.charAt(i));

}}}}}

**5.** **Method Overriding Parent:**

package Method\_Overriding;

public class MethodOverriding\_Parent {

public void java(String a, String b) {

System.***out***.println(a + " " + b);

}

public void program(int c, int d) {

System.***out***.println(c + "," + d);

}

public static void main(String[] args) {

MethodOverriding\_Parent obj = new MethodOverriding\_Parent();

obj.java("Codoid", "Solutions");

obj.program(5, 10);

}

}

**Method Overriding Child:**

package Method\_Overriding;

public class MethodOverriding\_Child extends MethodOverriding\_Parent {

*@Override*

public void java(String a, String b) {

super.java(a, b);

}

*@Override*

public void program(int c, int d) {

super.program(c, d);

}

public static void main(String[] args) {

MethodOverriding\_Child obj = new MethodOverriding\_Child();

obj.java("c", "sol");

obj.program(1, 2);

}

}

**6.** Some codes that may end up in an infinite loop

**Example:**

1. for(;;) {

System.out.println(“Infinite loop”);

}

1. int i=0;

while(i<5)

{

System.out.println(“Infinite loop”);

}

1. int i=0;

do

{

System.out.println(“Infinite loop”+i);

}

while(i<5)

**7. Method Overloading:**

package codoid.org;

public class MethodOverloading {

private void Program(int a, int b) {

System.*out*.println(a + b);

}

private void Program(int c, String d) {

System.*out*.println(c + "," + d);

}

private void Program(String e, String f) {

System.*out*.println(e + "," + f);

}

public static void main(String[] args) {

MethodOverloading obj = new MethodOverloading();

obj.Program(10, 20);

obj.Program(5, "Java");

obj.Program("JAVA", "LANGUAGE");

}

}

**8.Method Overloading by changing number of arguments:**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int add(int a, int b, int c) {

return a + b + c;

}

public static void main(String[] args) {

Calculator calc = new Calculator();

System.out.println(calc.add(1, 2));

System.out.println(calc.add(1, 2, 3));

}

}

**9.**We can read a file by Filereader, Bufferreader or Scanner.

**14.**Preserve spaces or other special characters, we can use the regular expression [^a-zA-Z0-9\s], which matches any character that is not a letter, digit, or whitespace character.

**15.**To remove duplicate elements from an array in Java, we can use a Set.

**20. Null pointer exception:**

public class QuestionNo\_20 {

public static void main(String[] args) {

String str = null;

System.*out*.println(str.length());

}}

**23.StringJoiner:**

* StringJoiner is a class which is used to construct a sequence of characters separated by a delimiter such as a comma, semicolon, or space.
* It provides an easy and efficient way to join multiple strings into a single string with a specific separator.

**25. Break and Continue:**

public class QuestionNo\_25 {

public static void main(String[] args) {

int i = 1;

while (i <= 10) {

if (i == 5) {

i++;

continue;

}

if (i == 8) {

break;

}

System.*out*.print(i + " ");

i++;

}

System.*out*.println("Done.");

}

}

**26. Program:**

package codoid.org;

public class QuestionNo\_26 {

public static void main(String[] args) {

for (int i = 1; i <= 7; i++) {

for (int j = 1; j <= i; j++) {

System.*out*.print(j);

}

System.*out*.print("\*");

}

}

}

**29.Multiple inheritance using Interface:**

**Multiple\_A:**

package QuestionNo\_29;

public interface Multiple\_A {

void car();

}

**Multiple\_B:**

public interface Multiple\_B {

void bike();

}

**Multiple\_C:**

public class Multiple\_C implements Multiple\_A, Multiple\_B {

*@Override*

public void bike() {

System.*out*.println("FZ");

}

*@Override*

public void car() {

System.*out*.println("PORSCHE");

}

public static void main(String[] args) {

Multiple\_C obj = new Multiple\_C();

obj.bike();

obj.car();

}

}

**31.Method Overloading** & **Method Overriding:**

**// method overloading**

package codoid.org;

public class Method\_Overloading {

public static int sum(int a, int b) {

return a + b;

}

public static int sum(int a, int b, int c) {

return a + b + c;

}

public static void main(String[] args) {

int result1 = *sum*(50, 50);

int result2 = *sum*(10, 20, 30);

System.*out*.println("result is" + result1);

System.*out*.println("result is" + result2);

}

}

**//Method Overriding**

package codoid.org;

public class Method\_Overriding {

public void Animals() {

System.*out*.println("Animals makes sound");

}

}

public class Method\_Overriding2 extends Method\_Overriding {

*@Override*

public void Animals() {

System.*out*.println("dog barks");

}

public static void main(String[] args) {

Method\_Overriding2 obj=new Method\_Overriding2();

obj.Animals();

}

}

**36.Fibonacci Series:**

package codoid.org;

public class QuestionNo\_36 {

public static void main(String[] args) {

int a = 0, b = 1, c;

System.*out*.println("Fibonacci series");

System.*out*.print(a + ",");

System.*out*.print(b + ",");

for (int i = 0; i <= 10; i++) {

c = a + b;

System.*out*.print(c + ",");

a = b;

b = c;

}

}

}

**37.String Buffer and String:**

**String Buffer:** String Buffer is a mutable class that allows you to modify the contents of a string without creating a new object.

**String**: String is an immutable class, which means that once a String object is created, its value cannot be changed.

**38. Constructor Overloading:**

Constructor overloading allows a class to have multiple constructors with different sets of parameters, which can be used to create objects of the class in different ways. Each constructor can have a different number and type of parameters.

**41. Swap two numbers without using temporary variable:**

package codoid.org;

import java.util.Scanner;

public class QuestionNo\_41 {

public static void main(String[] args) {

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

System.*out*.println("Enter your First number:");

int a = sc.nextInt();

System.*out*.println("first number is" + a);

System.*out*.println("Enter your Second number:");

int b = sc.nextInt();

System.*out*.println("first number is" + b);

a = a + b;

b = a - b;

a = a - b;

System.*out*.println("after swapping");

System.*out*.println(a);

System.*out*.println(b);

}

}

**57.** **Convert string to char and vice versa**

package codoid.org;

public class QuestionNo\_57 {

public static void main(String[] args) {

String a = "JAVA";

String[] split = a.split("");

for (String string : split) {

System.*out*.println(string);

}

String[] split2 = a.split(" ");

for (String string : split2) {

System.*out*.println(string);

}

}

}

**59.Abstraction:**

**//Parent:**

package QuestionNo\_59;

public interface Abstraction\_Parent {

void car();

void bike();

void cycle();

}

**//Child:**

package QuestionNo\_59;

public class Abstraction\_Child implements Abstraction\_Parent {

*@Override*

public void car() {

System.***out***.println("audi");

}

*@Override*

public void bike() {

System.***out***.println("bmw");

}

*@Override*

public void cycle() {

System.***out***.println("ladybird");

}

public static void main(String[] args) {

Abstraction\_Parent obj = new Abstraction\_Child();

obj.car();

obj.bike();

obj.cycle();

}

}

**64. Nested If-else clause:**

int score = 85;

if (score >= 90) {

System.out.println("A");

}

else {

if (score >= 80) {

System.out.println("B");

}

else {

if (score >= 70) {

System.out.println("C");

}

else {

System.out.println("F");

}

}

}

**65.** **Odd or Even:**

package codoid.org;

import java.util.Scanner;

public class QuestionNo\_65

public static void main(String[] args) {

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

System.***out***.println("Enter your number:");

int n = sc.nextInt();

if (n % 2 == 0) {

System.***out***.println("The number is even");

} else {

System.***out***.println("The number is odd");}}}

**66. Multiplication Program:**

package codoid.org;

import java.util.Scanner;

public class QuestionNo\_66 {

public static void main(String[] args) {

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

System.***out***.println("Enter table to multiple");

int a = sc.nextInt();

System.***out***.println("Till which number to multiple");

int b = sc.nextInt();

int c;

for (int i = 0; i <= b; i++) {

c = a \* i;

System.***out***.println(a + "\*" + i + "=" + c);

}

}

}

**67. Armstrong or not:**

package codoid.org;

import java.util.Scanner;

public class QuestionN0\_67 {

public static void main(String[] args) {

Scanner s = new Scanner(System.***in***);

System.***out***.println("Enter Your Number:");

int n = s.nextInt();

int arm = 0, rem = 0, i = n;

while (n > 0) {

rem = n % 10;

arm = arm + (rem \* rem \* rem);

n = n / 10;

}

if (arm == i) {

System.***out***.println("Armstrong number");

}

else {

System.***out***.println("Not an armstrong number");

}

}

}

**80**. public class Factorial {

public static void main(String[] args) {

int num = 5;

int factorial = findFactorial(num);

System.out.println("Factorial of " + num + " is " + factorial);

}

public static int findFactorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* findFactorial(n-1);

}

}

}

\*\*\*\*\*

**SQL**

1. **Scalar functions**:

Scalarfunctions that take one or more input parameters and return a single value. They can be used in various SQL statements, including SELECT, WHERE, and ORDER BY clauses.

Ex:

SELECT UPPER('hello world')

The above query will return 'HELLO WORLD' as output because the UPPER() function converts all the characters in the input string to uppercase.

**2.** **Joins:**

Joins is used to combine data from two or more tables in SQL.

Types:

Inner join left outer join, right outer join, and full outer join.

Example of an INNER JOIN:

SELECT \*FROM customers INNER JOIN orders ON customers.customer\_id = orders.customer\_id

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **customer\_id** | **customer\_name** | **order\_id** | **order\_date** | **amount** |
| 1 | X | 1234 | 01-01-2023 | 50 |
| 2 | Y | 1235 | 02-02-2023 | 100 |

**3. Rename Column Name:**

Rename a column name in SQL, we can use the ALTER TABLE statement with the RENAME COLUMN clause.

Example:

ALTER TABLE employees RENAME COLUMN emp\_name TO employee\_name;

The above query will rename the emp\_name column in the employees table to employee\_name.

**5.The Distinct Statement** :

Distinct Statement is used in SQL to remove duplicate rows from the result set of a query. It only returns the unique values in the specified columns.

Syntax:

SELECT DISTINCT column1, column2, ...FROM table\_name;

The DISTINCT statement is useful in scenarios where we want to get a list of unique values from a table, without duplicates.

**6. Maximum salary:**

Print the maximum salary for a particular department in SQL, we can use the MAX() function and the WHERE clause to filter the result set by department.

Example:

SELECT MAX(salary) FROM employees WHERE department = 'Sales';

7. **Different operators in SQL:**

There are various operators in SQL, including arithmetic operators, comparison operators, logical operators, and string operators.

Arithmetic operators:

* Addition: +
* Subtraction: -
* Multiplication: \*
* Division: /
* Modulus: %

Comparison operators:

* Equal to: =
* Not equal to: !=
* Greater than: >
* Less than: <
* Greater than or equal to: >=
* Less than or equal to: <=

Logical operators:

* AND
* OR
* NOT

String operators:

* Concatenation
* Like
* Not like

8. **Query to display first 5 records from the employee table:**

We can use the LIMIT clause in SQL. The LIMIT clause is used to limit the number of rows returned by a query.

Example:

SELECT \* FROM employee LIMIT 5;

9. **Query to display last 5 records from the employee table:**

We can use the ORDER BY and DESC clauses in SQL. The ORDER BY clause is used to sort the result set by a column, and the DESC clause is used to sort the result set in descending order.

Example:

SELECT \* FROM employee ORDER BY emp\_id DESC LIMIT 5;

\*\*\*\*\*

**MT**

1.**Bug report for a bug in whatsapp:**

**Summary:** Unable to send video files in WhatsApp group chat

**Severity:** Major

**Priority:** High

**Description:** The user is unable to send a video file in WhatsApp group chat. The user selects a video from the gallery and clicks on the send button but the file fails to upload and an error message appears.

**Steps to Reproduce:**

1. Launch the WhatsApp application.
2. Tap on the group .
3. Click + icon to send a file
4. Select video from the gallery
5. Click on the send button
6. Observe that video fails to upload and error message occurs.

**Expected Result:** After selecting a video file and hitting the send button, the video file should be uploaded and sent to the group chat.

**Actual Result:** After selecting a video file and hitting the send button, the video fails to upload and the error message pops up.

2.**Bug report for a login page**

**Summary:** Submit button is not working in a login page

**Severity:** Major

**Priority:** High

**Description:** The Submit button is not working in a login page. The user enters their username and password but when they click on submit button it is unable to go to homepage.

**Steps to Reproduce:**

1. Launch the application
2. Enter the valid username
3. Enter the valid password
4. Click on the submit button
5. Observe that submit button is not working

**Expected Result:** After entering all the credentials and by clicking the submit button it should be logged in and navigate to the homepage.

**Actual Result:** After clicking the submit button it is not navigating to the home page and the user is not logged into the system.

**3. Bug report for whatsapp**

**Summary:** Unable to send message in whatsapp

**Severity:** Major

**Priority:** High

**Description:** The user launches the whatsapp application. The user opens the chat box and types a message. After typing a message user hits on a send button, the message does not get delivered to the receiver.

**Steps to Reproduce:**

1. Launch the application
2. Select a contact
3. Opens a chat box
4. Type a message
5. Click on send button
6. Observe that message is not delivered.

**Expected Result:** After entering all the credentials and by clicking the submit button it should be logged in and navigate to the homepage.

**Actual Result:** After clicking the submit button it is not navigating to the home page and the user is not logged into the system.

**5**.**Bug report for whatsapp profile picture:**

**Summary:** Profile Picture not updated in WhatsApp

**Severity:** Major

**Priority:** High

**Description:** The user is unable to update their profile picture in WhatsApp. After selecting a new picture from the gallery or taking a new picture with the camera, the application does not show the new picture in the profile. The user has tried multiple times to update the profile picture, but the old picture remains unchanged.

**Steps to Reproduce:**

1. Launch the WhatsApp application on the mobile device.
2. Tap on the user profile picture in the top left corner of the screen.
3. Tap on the "Edit" button next to the profile picture.
4. Select a new picture from the gallery or take a new picture with the camera.
5. Adjust the picture as necessary and tap on the "Save" button.
6. Observe that the old profile picture is still displayed.

**Expected Result:** After selecting a new picture and saving it, the new profile picture should be displayed.

**Actual Result:** The old profile picture remains unchanged after selecting and saving a new picture.

**For the Question 8, 9. Refer the below sheet.**

