Interview Preparation

Basics Programm.

1) max out of 3 Int
→ int a, b, c
if (a>b && b>c)
{ max = a }
else if (b>a && b>c)
{ max = b }
else
{ max = c }

2) Swap 2 No.
int n1, n2; 100, 50
n1 = n1+n2 n1 = 100+50
n2 = n1-n2 n2 = 150-50 = 100
n1 = n1-n2 n1 = 150-100 = 50

3) Random 5 Int.
for (int i=1; i>n; i++)
{
Syso (int (math.Random (2*50)))
}

4) Leap year
if (y % 400 = 0) { 400% T, 100% F, 4 T }
{
flag = T
}
else if (y % 100 = 0)
{
flag = F
}
else if (y % 4 = 0)
{
flag = T
}
else
{
flag F
}
if (flag = T) = Leap
else (flag = F) = Not

5) Sum of n no.
int n, sum = 0
for (int i=1; i<=n; i++)
{ sum = sum + i; }

6) Even odd
int n.
if (n % 2 == 0)
{ even }
{ odd }

7) fibonacci Series
P1 = 0, P2 = 1, P3 = 0
int n = 5.
for (i=1; i<=n; i++)
{
Syso (P1)
P3 = P1 + P2
P1 = P2
P2 = P3
}

n = 5

1st	2nd	3rd
0	0,1	0,1,1
P3 = 0+1 = 1	P3 = 1+1	P3 = 1+2
P1 = 1	P1 = 1	P1 = 2
P2 = 1	P2 = 2	P2 = 3

4th	5th
0,1,1,2	0,1,1,2,3
P3 = 2+3	P3 = 3+5
P1 = 3	P1 = 5
P2 = 5	P2 = 8

8) Reverse
n = 536, Rev = 0, d = 0;
while (n>0)
{ Rev = Rev * 10;
d = n % 10;
Rev = Rev + d;
n = n / 10;
}
Syso (Rev)

n = 536.

Rev = 0*10 = 0	Rev = 6*10 = 60	Rev = 63*10 = 630
d = 536 % 10 = 6	d = 53 % 10 = 3	d = 5 % 10 = 0.5 = 5
Rev = 0+6 = 6	Rev = 60+3 = 63	Rev = 630+5 = 635
n = 536/10 = 53	n = 53/10 = 5	n = 5/10 = 0.5 = 0

9) Prime No.
n = 7, count
For (i=1; i<=n; i++)
{
if (n % i == 0)
{
count ++;
}
if (count == 2)
{
Prime
}
else { Not Prime }

10) Sum of digit
n = 536, sum, d.
while (n>0)
{ d = n % 10
sum = sum + d;
n = n / 10
}

11) Armstrong
n = 534, sum, d
while (n>0)
{
d = n % 10
sum = sum + (d*d*d)
n = n / 10;
}
if sum = n ⇒ Armstrong
else Not

12) String Reverse
String Str = in.nextLine();
n = Str.length();
for (int i = n-1; i>=0; i--)
{
Rev = Rev + charAt(i);
}
Syso (Rev)

Palindrome ?
if (Str == Rev)
{ palindrome }
else { Not palindrome }

Str = Ashish (1 2 3 4 5 6) length = 6
(0 1 2 3 4 5) ← charAt(i)

Scanned by CamScanner

4 pillers in java.

4) Interface (100% abstraction)

i) support multiple inheritance
ii) Public Static final
iii) public & Abstract method.
iv) class implements interface & extends class.
v) interface cannot implement interface.
vi) interface can extend interface one or more.

```
Interface A
{ int a;
  void P();
}

Interface C extends A
{ void P1()
}

class B implements C
{
  void P();  void P1()
  { }        { }
  psvm()
  { B b = new B();
    b.P();
    b.P1();
  }
}
```

Datatype

Primitive	Non-Primitive
i) byte ← (8)	Array
ii) short ← (16) Integer	String
iii) int ← 32 bit	class
iv) Long ← 64	Object
v) float (32) Float	
vi) double (64)	
vii) boolean ← 16	
viii) character ← 16	

Heap memory

Stack memory

Array.

i) `int a[] = new int[n]`
`int a[][] = new int[][]`

```
class A
{ psvm()
  { int a[] = new int[5];
    a[] = { 1,2,3,4,5 }
    int large = a[0];
    for (int i=0; i<a[].length; i++)
    {
      if (a[i] > large)
      { large = a[i];
      }
      sysO (large = " + a[i]);
```

Sort.

```
for (i=0; a i<a.length
      i) ; i++)
{
  for (j=i+1 = j<a.length(); j++)
    if (a[i] > a[j])
    {
      temp = a[i]
      a[i] = a[j]
      a[j] = temp;
    }
}
```

1) Polymorphism

i) Overloading
ii) Overriding.

i) Overloading →

```
class A
{ m1(int a, int b)
  m2(int a, int b, int c)
  psvm()
  { A a = new A();
    a.m1(1,2);
    a.m2(1,2,3);
  }
}
```

Static binding

Overriding is expressing inheritance.

ii) Overloading

```
class A {
  m1(int a, int b)
}
class B extends A
{
  m1(int a, int b)
  psvm()
  { B b = new B();
    b.m1();
    A a = new A();
    a.m1();
  }
}
```

- dynamic binding.
- Subclass object can have properties of superclass. Not vice versa.

2) Abstraction (Hiding Implementation)

i) Partial abstraction
ii) Concrete Methods have.

```
Abstract class A
{ m1()
  { }
  public void abstract m2();
}
class B extends A
{
  public void m2()
  { }
  Public void m3()
  { }
  psvm()
  { A a = new B()
    a.m1()
    a.m2()
```

Type

i) Single, Multiple-level, Hierarchical

ii) Multiple — not possible

Scanned by CamScanner

Selenium :

- To bypass firewall.
  firefox Profile f = new ProfileIni().getProfile ("default");
  WebDriver w = new FirefoxDriver(f);

Initialize
WebDriver w = new FirefoxDriver()
WebDriver wI = new ~~FGetDriver~~ InternetExplorerDriver();
WebDriver we = new ChromeDriver();

Locators :- Id, Name, Link Text, TagName, ClassName, xpath;

- Cross Browser
  System.setProperty ("webdriver.ie.driver", "path.exe");

- Selection
  Select s = new Select (w.findElement(By.cssName("--")));
  s.selectByValue("");
  s.selectByIndex(num);
  s.selectByVisibleText("");

- Listing
  WebDriver w = new FirefoxDriver();
  (Interface) WebElement we = w.findElement(By.name(""));
  List l = we.findElement(By.tagname("a"));
  (common locator in table to obtain list)

- Popup :-
  WebDriver w = new FirefoxDriver();
  (Interface) Alert a = w.switchTo().alert();
  a.accept(); a.dismiss();

- Navigate
  Navigate().To("URL");
  Navigate.back();
  Navigate.forward();

- Action (move hovering)
  ~~FirefoxDriver~~ WebDriver w = new ~~WebDriver~~
  Action a = new Action(w);
  ~~WebElement we = new WebElement(By.find);~~
  WebElement we = w.findElement(By.name(""));
  a.moveToElement(we);
  a.click.build.perform();

- Upload file (SendKeys)
  w.findElement By.xpath("").sendKeys("path");

- Screen shot :-
  File f = ((TakeScreenShot) driver.getScreenShot As (OutputType.~~FILE~~ FILE);
  FileUtils.copyFile(f, new File("Path"));

- Press keys
  Single (Enter, tab, shift)
  W.findElement(By.name()).sendKeys(Keys.ENTER);

  Shift + tab
  String keys = Keys.chord(Keys.SHIFT, Keys.ENTER);
  w.findElement(By.name("")).sendKeys(keys);

- maximize browser
  w.manage.window.maximize();

- Radio buttons
  i) make list
  ii) check size
  iii) click button
  iv) check selected or not
  List l = w.findElement(By.name(""));
  int s = l.size();
  w.findElement(By.id("")).click();
  boolean b = w.findElement(By.id()).isSelected();

Scanned by CamScanner

What is the use of getOptions() method?

What is the difference between findElement() and findElements()?

findElement( ):

1. Find the first element within the current page using the locators.
2. Returns a single WebElement.
3. If element not found it will throw exception NoSuchElementException.

findElements( ):

1. Find all elements within the current page using the locators.
2. Returns List of WebElements.
3. If element not found it will return empty List of WebElement object.

What is the difference between driver.close() and driver.quit()?

How to capture screenshot in WebDriver?

How to capture screenshot in WebDriver for

Puma® Tees, Jackets, Pants
PUMA Store

What are the limitations of Selenium?

Technical challenges with Selenium are:

- Selenium supports only web based applications.
- It does not support the Bitmap comparison.
- For any reporting related capabilities have to depend on third party tools.
- No vendor support for tool compared to commercial tools like QTP/UFT.
- As there is no object repository concept in Selenium, maintainability of objects becomes difficult.

What is WebDriver backed selenium?

What is the difference between "/" and "//" in

What is Absolute and Relative xPath?

Absolute xPath: Writing the complete path of a specified element using single forward slash ('/') is known as Absolute xPath. It is time consuming and lengthy. Here the single slash ('/') represents immediate child.

Example: html/body/div[7]/div[3]/div/div[2]/div[1]/span

Relative xPath: Writing the path of a specific element using double forward slash ('//') is known as Relative xPath. Here the double slash ('//') represents any child also called as Descendants.

Example: //table[@class='dataTable']/tbody/tr[1]

Scanned by CamScanner

driver.get():

1. driver.get() method is generally used to open the url.
2. It will till the whole page gets loaded.

driver.navigate():

1. driver.navigate() method is generally used to navigate to url.
2. It supports back, forward, and page refresh.

What is the difference between driver.close() and driver.quit()?

close(): close() method closes the web browser window that the user is currently working on.

quit(): whereas quit() method closes down all the windows that the program has opened.

Scanned by CamScanner

How do you handle radio buttons?

```
WebDriver driver = new FirefoxDriver();
driver.get("http://www.xyz.com");

//radio button can find by using name attribute
List element = driver.findElements(By.name("sex"));

//by using size() method, get the count of radio buttons
int btnCount = element.size();
System.out.println(btnCount);

//to click the radio button
driver.findElement(By.id("u_0_e")).click();

//check whether the button is selected or not
// returns true if selected else false
boolean btn1 = driver.findElement(By.id("u_0_e")).isSelected();
System.out.println(btn1);
```

Scanned by CamScanner

How do you send ENTER Key?

```
WebDriver driver=new FirefoxDriver();
driver.get("https://www.xyz.com");

//Enter Key
driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);

// Tab Key
driver.findElement(By.xpath("xpath")).sendKeys(Keys.TAB);
```

How to capture screenshot in WebDriver?

The sample code to take screenshot of webpage is:

```
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("D:\\Test.jpg"));
```

Scanned by CamScanner

Write a code to print the number of links in a page?

```
public class Link_Count{

 public static void main(String[] args){

  WebDriver driver=new FirefoxDriver();
  driver.get("http://www.google.com");

  //find links by anchor tag
  List element=driver.findElements(By.tagName("a"));
  //print the number of links
  System.out.println(element.size());

 }
}
```

Scanned by CamScanner

What are the different types Waits in WebDriver?

Waiting provides some time interval between actions performed; mostly locating an element or any other operations to be performed on the element.

Implicit Wait:

Implicit wait tells Selenium instance to wait for specified amount of time before throwing an exception. If WebDriver cannot find an element, it will wait for specified amount of time. During this time, no attempt is made to find the element. After completion of specified time, WebDriver will try to find the element. If the element is not found then we get an exception.

Implicit wait can be implemented by using the code below:

```
driver.manage().timeouts().implicitlyWait(10,
```

Explicit Wait:

You can define to wait for a certain condition to occur before proceeding further test code execution.

Explicit wait can be implemented by using the code below:

```
WebDriverWait wait = new WebDriverWait(driver,
wait.until(ExpectedConditions.elementToBeClick
```

Fluent Wait:

Each FluentWait instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. And you can ignore specific types of exceptions whilst waiting, such as NoSuchElement Exceptions when searching for an element on the page.

Fluent wait can be implemented by using the code below:

```
Wait wait = new FluentWait(driver)
  .withTimeout(30, SECONDS)
  .pollingEvery(5, SECONDS)
  .ignoring(NoSuchElementException.class);
```

Scanned by CamScanner

How to refresh a web page using WebDriver?

The following are the various methods available in WebDriver to refresh the page:

1. Using refresh() method:

```
driver.get("https://google.com/");
driver.navigate().refresh();
```

2. Using to() method:

```
driver.get("https://google.com/");
driver.navigate().to(driver.getCurrentUrl());
```

3. Using get() method:

```
driver.get("https://google.com/");
driver.get(driver.getCurrentUrl());
```

4. Using SendKeys method:

```
t("https://google.com/");
ndElement(By.xpath("xpath")).sendKeys(Keys.F5)
```

Scanned by CamScanner

What are the different ways in which you can produce reports for TestNG results?

There are two ways to produce a report with TestNG, they are:

- **Listeners:** For a listener class to implement, the class has to implement the org.testng.TestListener interface. These classes are informed at runtime by TestNG when the test begins, finishes, skips, passes or fails.
- **Reporters:** For a reporting class to implement, the class has to implement an org.testng.Reporter interface. When the whole suite run ends, these classes are called. When called, the object consisting the information of the whole test run is delivered to this class.

Scanned by CamScanner

What is @DataProvider annotation?

@DataProvider: Marks a method as supplying data for a test method. The annotated method must return an Object[][] where each Object[] can be assigned the parameter list of the test method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.

```
public class DataProviderExample{

 @Test(dataProvider="getData")
 public void setData(String username, String p

  System.out.println("Username: "+username);
  System.out.println("PWD:"+password);
 }

 @DataProvider
 public Object[][] getData(){
 //Rows - No of test iteration.
 //Columns - No of parameters in test data.
 Object[][] data=new Object[3][2];

 // 1st row
 data[0][0]="user1";
 data[0][1]="pwd1";

 // 2nd row
 data[1][0]="user2";
 data[1][1]="pwd2";

 // 3rd row
 data[2][0]="user3";
 data[2][1]="pwd3";

 return data;
```

Scanned by CamScanner

What is timeout in TestNG?

The maximum number of milliseconds the test should take.

Sample Code:

```
@Test(timeOut=1000) // time in milliseconds
public void timeOutExample(){
  System.out.println("Timeout test");
}
```

The above test should be executed within 1000 milliseconds. Other test will be failed.

Scanned by CamScanner

How do you skip a test in TestNG?

@Test(enabled=false) annotation is used to skip a test case if it is not ready to test.

Scanned by CamScanner

What is expected annotation?

We use "expected" with @Test annotation and specify the type of exceptions that are expected to be thrown when executing the test methods.

The below example which throws an exception called “ArithmeticException” when dividing two numbers with denominator value as 0. The test will pass since we handled the exception by “expected” annotation.

```
@Test(expected=ArithmeticException.class)
public void dividedByZeroEx(){
  int e=1/0;
}
```

When you run the test without “(expected = ArithmeticException.class)”, then it will throw the exception “java.lang.ArithmeticException: / by zero”. And the test will fail.

```
@Test
public void dividedByZeroExample2(){
  int e=1/0;
}
```

Scanned by CamScanner

What is syntax to set test method dependency for multiple test methods?

We can set test method's dependency on multiple test methods as bellow.

```
@Test(dependsOnMethods={"Login","checkMail"})
public void Logout(){
  System.out.println("Logout Test.");
}
```

We set dependency on @Test methods to skip its execution if the depending on method fails.

Scanned by CamScanner

How do you execute group test cases in TestNG?

We can execute only set of group and exclude another set. This gives us the maximum flexibility in divide tests and doesn't require us to recompile anything if you want to run two different set of tests back to back. Groups are specified in testng.xml file.

Sample Code:

```
public class groupExamples{

 @Test(groups="Regression")
 public void testCaseOne(){
  System.out.println("Regression Group");
 }

 @Test(groups="Regression")
 public void testCaseTwo(){
  System.out.println("Regression Group");
 }

 @Test(groups="Retest")
 public void testCaseThree(){
  System.out.println("Retest Group");
 }

 @Test(groups="Regression")
 public void testCaseFour(){
  System.out.println("Retest Group");
 }
}
```

Scanned by CamScanner

How to use assertions in TestNG explain with one example?

There are many different assertions available In TestNG but few important assertions are:

1. assertEquals
2. assertNotEquals
3. assertTrue
4. assertFalse
5. assertNull
6. assertNotNull

Assert.assertEquals(actual, expected):
This assertion is used to compare expected and actual values in selenium WebDriver. If both values are same then it will continue execution. But if fails then immediately it will mark the test method as fail and exit from that test method.

Sample Code:

```
@Test
 public void assertCheck(){
  WebDriver driver=new FirefoxDriver();
  driver.navigate().to("http://gmail.com");
  Assert.assertEquals("Gmail", driver.getTitle
 }
```

The above test will pass since both value matches.

Scanned by CamScanner

```
22      void getapi()
23      {
24
25          ExtentHtmlReporter extent = new
    ExtentHtmlReporter(new File(System
    .getProperty("user.dir")+"/Reports/Api.html"
    ));
26          ExtentReports reports = new
    ExtentReports();
27          reports.attachReporter(extent);
28          ExtentTest logger;
29          logger= reports.createTest("API
    test");
30          logger.info("Starting Api getting
    info")
31
32          //URI
33          RestAssured.baseURI ="http://restapi
    .demoqa.com/utilities/weather/city";
34
35          //Resuestobject
36          RequestSpecification httpreq=
    RestAssured.given();
37
38          //response obj
39          Response resp =httpreq.request
    (Method.GET,"/Hyderabad");
40
41          String a =resp.getBody().asString();
42          System.out.println("response body "
    +a);
43
44  File src = ((TakesScreenshot)w
    ).getScreenshotAs(OutputType.FILE);
45          FileHandler.copy(src, new File
    (System.getProperty("user.dir")+"/Screenshot
    /login.png"));
46
47          logger.pass("End info got");
48          reports.flush();
49      }
50
```

Scanned by CamScanner