1. What is the difference between quit() and close() ?
2. quit() shall quit the driver and close() shall close the current window.
3. quit() shall quit the browser of the active window and close() shall close all the open browsers
4. quit() shall quit the driver and close() shall close all the open browsers
5. None of these
6. Identify the incorrect locator identifier supported by Selenium WebDriver.
7. ClassName and Name
8. Div and Span
9. Xpath and CSS
10. LinkText and PartialLinkText
11. Identify the correct syntax for customized CSS and Xpath supported by Selenium WebDriver.
12. CSS - tagname.classname, Xpath - //[@attribute='value'].
13. CSS - tagName#id, Xpath - //tagName[attribute=”value”].
14. CSS - tagName#, Xpath - //\*[@attribute='value'].
15. CSS - tagName[attribute='value'], Xpath - //tagName[@attribute='value'].
16. Identify of the xpath from the html code for the second element of the unordered list (<li name=”item2”>Css</li>, assuming that the parent (div tag) is having a dynamic value for locator Id.

<div id=”name\_12”>

<ul>

<li name=”item1”>Xpath</li>

<li name=”item2”>Css</li>

<li name=”item3”>Partial Text</li>

</ul>

</div>

1. //li[@name=’item1’]/following-sibling::li
2. //li[@name=’item1’]/li[1]
3. //li[@name=’item1’]/following-sibling::li[2]
4. //ul[@name=’item1’]/li/li
5. How do count number of checkboxes on a webpage?
6. driver.findElements(By.tagname(“a”)).size()
7. driver.findElements(By.tagname(“input”)).length()
8. driver.findElements(By.cssSelector(“input[type=’checkbox’]”)).length()
9. driver.findElements(By.cssSelector(“input[type=’checkbox’]”)).size()
10. Identify the method used in Selenium WebDriver to verify the presence of a web element within the web page.
11. isChecked()
12. isDisplayed()
13. isPresent()
14. isEnabled()
15. Identify the correct syntax for declaring implicit wait in Selenium WebDriver.
16. driver.manage().implictWait(5,TimeUnit.SECONDS)
17. driver.manage().timeouts().implictlyWait(5,TimeUnit.MILLISECONDS)
18. driver.manage().timeouts().implictlyWait(5,SECONDS)
19. driver.manage().timeouts().implictWait(5,TimeUnit. MILLISECONDS)
20. Which class is responsible to achieve explicit wait in Selenium WebDriver?
21. WebDriver wait = new WebDriver()
22. Wait w = new Wait(driver,4)
23. WebDriverWait w = new WebDriverWait(driver,5)
24. WebBrowserWait w = new WebBrowserWait(driver,5)
25. How to perform right click in Selenium?
26. Actions a = new Actions(driver)

a.moveToElement (driver.findElement(By.id(“name”))).rightClick().build().perform()

1. Actions a = new Actions()

a.moveToElement (driver.findElement(By.id(“name”))).contextClick().build().perform()

1. Action a = new Action(driver)

a.moveToElement (driver.findElement(By.id(“name”))).contextClick().build().perform()

1. Actions a = new Actions(driver)

a.moveToElement (driver.findElement(By.id(“name”))).contextClick().perform()

1. How to navigate to the parent window when there are multiple windows?
2. Set<String> childWindows = driver.getWindowHandles();

Iterator<String> iterator = childWindows.iterator();

String parent = iterator.next();

driver.switchTo(parent);

1. Set<String> childWindows = driver.getWindowHandles();

Iterator<String> iterator = childWindows.iterator();

String parent = iterator.next();

driver.switchTo().window(parent);

1. childWindows = driver.getWindowHandles();

Iterator<String> iterator = childWindows.iterator();

String parent = iterator.next();

driver.switchTo().window(parent);

1. Set<String> childWindows = driver.getWindowsHandles();

Iterator<String> iterator = childWindows.iterator();

String parent = iterator.next();

driver.switchTo().window(parent);

1. How to capture screenshots with Selenium Web driver?
2. File s = driver.getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(s, new File (“C:\\Users\\Screenshots\\screenshot.png”));

1. File s = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(s, new File (“C:\\Users\\Screenshots\\screenshot.png”));

1. File s = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(new File (“C:\\Users\\Screenshots\\screenshot.png”));

1. File s = ((TakesScreenshot)driver).getScreenshotAs();

FileUtils.copyFile(s, new File (“C:\\Users\\Screenshots\\screenshot.png”));

1. Study the following the block of xml code then identify correct statement.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Test Cycle">

<test name = “Regression”>

<classes>

<class name=”test.cycle1”/>

<methods>

<include name =”login.\*”/>

<exclude name =”validateDate.\*”/>

</methods>

</classes>

</test>

</suite>

1. Test methods staring with ‘login’ and ‘validateDate’ will be executed.
2. Only ‘login’ test method will not be executed.
3. Test methods staring with ‘validateDate’ will not be executed.
4. Only ‘validateDate’ test method will not be executed.
5. Identify the set of test cases that will only be triggered from a suite from the following block of xml code.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Test Cycle">

<test name = “Regression”>

<groups>

<run>

<include name =”Login”/>

</run>

</groups>

<classes>

<class name=”test.cycle1”/>

<class name=”test.cycle2”/>

<methods>

<include name =”search”/>

</methods>

</classes>

</test>

</suite>

1. Entire regression suite will be triggered for execution.
2. Test methods with tag ‘Login’ will be triggered for execution.
3. Test method ‘search’ will be triggered for execution.
4. All of these.
5. Which helper attribute skips a particular test from execution in TestNG?
6. dependsOnMethods
7. timeOut
8. enabled
9. exclude
10. Which plugin helps to execute test cases from test folder in Maven project?
11. Maven- invoker plugin
12. Maven – verifier plugin
13. Surefire plugin
14. None of these
15. Once Maven reads the pom.xml, first were does it search for the dependencies?
16. Central Repository
17. Local Repository
18. Remote Repository
19. None of these
20. What is the output of the following program?

class Derived

{

protected final void getDetails()

{

System.out.println("Derived class");

}

}

public class Test extends Derived

{

protected final void getDetails()

{

System.out.println("Test class");

}

public static void main(String[] args)

{

Derived obj = new Derived();

obj.getDetails();

}

}

1. Derived class
2. Test class
3. Runtime error
4. Compilation error
5. What is the output of the following program?

class Vehicle

{

protected void getData()

{

System.out.println("Inside Vehicle");

}

}

class Hyundai extends Vehicle

{

protected void getData()

{

System.out.println("Inside Hyundai");

}

}

public class Test

{

public static void main(String[] args)

{

Vehicle obj = new Hyundai();

obj.getData();

}

}

1. Inside Vehicle
2. Inside Hyundai
3. Compilation error
4. Runtime error
5. What is the output of the following program?

public class test

{

public static void main(String[] args)

{

List<String> list = new LinkedList<>();

list.add("a");

list.add("b");

list.add("c");

list.add("d");

Iterator<Integer> iter = list.iterator();

while (iter.hasNext())

System.out.printf(iter.next() + " ");

System.out.println();

}

}

1. abcd
2. Runtime error
3. Compilation error
4. How to handle exceptions in Selenium
5. Using try/catch
6. Using throws declaration
7. Using throw keyword
8. All the above