1.System.setProperty(“\_\_\_\_\_\_\_\_\_\_\_\_\_”,”Location of Driver”) for Internet Explorer Browser

a.WebDriver.IE.Driver

b.webdriver.ie.driver //

c.webdriver.internetexplorer.driver

d.none of above

2.Which Selenium command(s) simulates selecting a link?

a. findElement(By.Link("LinkName"))

b. findElement(By.Id("LinkName"))

c. findElement(By.linkText("LinkName")) //

d. findElement(By.xpath("LinkName"))

e. findElement(By.tagName("LinkName"))

3.What does the below lines of code meant for?

WebDriver driver = new FirefoxDriver();

Selenium selenium = new WebDriverBackedSelenium(driver, "http://www.techbeamers.com/");

a. Will simply start the browser and open the site.

b. Will run webdriver tests using selenium.

c. Will run Selenium 1.0 tests in webdriver. //

d. None of these

4.Which of the following statement describes the most appropriate way to delete the cookies?

a. We can delete cookies in 3 ways. //

driver.manage().deleteCookieNamed("CookieName");

driver.manage().deleteCookie(CookieId);

driver.manage().deleteAllCookies();

b. We can delete cookies in 2 ways.

driver.manage().deleteCookieNamed("CookieName");

driver.manage().deleteCookie(CookieId);

c. driver.manage().deleteCookie(CookieId);

d. driver.manage().deleteCookieNamed("CookieName");

5.Which of the following is a correct difference b/w getWindowHandles() and getWindowHandle()?

a. getWindowHandles() returns address of all the open browsers whereas getWindowHandle() gets the address of the current browser. //

b. getWindowHandles() returns a String whereas getWindowHandle() returns an Iterator<String>

c. getWindowHandles() returns the active browser handle whereas getWindowHandle() gives the top most browser handle

d. None of these.

6. Which method can be used as alternative of get method in WebDriver to open URL in browser

a. goToUrl("url")

b. navigate().to("URL") //

c. getUrl("url")

d. getTo("url")

7.How to get all the links present on the page?

a. List < WebElement > alllinks =d.findElements(By.xpath("//a"));

b. List < WebElement > alllinks =d.findElements(By.tagName("a"));

c. All of the option

d. None of the option .

8.What is the purpose of WebDriverBackedSelenium class in WebDriver application?

a. It's a Selenium class to call WebDriver APIs.

b. It's a generic implementation of WebDriver to support multiple browsers.

c. It's a WebDriver class implementing the Selenium-RC APIs. //

d. None of these

9.What does the below Selenium WebDriver code would subject to do?

Alert alert = driver.switchTo().dismiss();

a. This will close all of the currently open alert popups.

b. This will close the currently open alert popup. //

c. This will return the currently open alert popup.

d. This will create a new alert popup.

10.Cron pattern for running script at 20:15 (08:15 PM) every Saturday

1.15 08 \* \* 6

2.15 20 \* \* 6

11.Cron

12.What will be returned for an Element not found by findElement() method in selenium?

a. Return Empty List.

b. Returns null

c. It doesn't throw any exception.

d. It throws NoSuchElementException. //

13.Which class is for double click on element or use any key from keyboard using webdriver?

a. Select class

b. Actions class 🡨

c. Both of the above.

d. None of the above.

**14.Which of the way is not for scrolling using WebDriver ?**  
1.((JavascriptExecutor) driver).executeScript(“arguments[0].scrollIntoView();”, element);  
2.((JavascriptExecutor) driver).executeScript(“window.scrollBy(x-pixels,y-pixels)”);  
3.((JavascriptExecutor) driver).executeScript("window.scrollBy(0, document.body.scrollHeight)");

4. ((JavascriptExecutor) driver).executeScript(“arguments.scrollIntoView(true);”, element);

15.Arrange Execution Flow of Below Junit 5 Annotation

a.BeforeEach

b.BeforeAll

c.Test

d.AfterAll

e.AfterEach

1. BeforeEach > BeforeAll > Test > AfterEach > AfterAll

2. BeforeAll > BeforeEach > Test > AfterEach > AfterAll

3. BeforeAll > AfterAll > Test > AfterEach > BeforeEach

3. Test > AfterAll > BeforeAll > AfterEach > BeforeEach

16. **In following code which exception will occur**

**public** **class** Demo

{

WebDriver driver;

@Test

**public** **void** f1()

{

System.*setProperty*("webdriver.chrome.driver", "path of chromedriver");

WebDriver driver=**new** ChromeDriver();

driver.get("http://www.google.com");

}

}

1.FileNotFoundException

2.NullPointerException //

3.MalformedURLException

4.No Exception

17. **In following code which exception will occur**

{ WebDriver driver;

@Test

**public** **void** f1()

{

System.*setProperty*("webdriver.chrome.driver", "path of chromedriver");

driver=**new** ChromeDriver();

driver.get("www.google.com");

}

}

1.FileNotFoundException

2.NullPointerException

3.MalformedURLException //

4.No Exception

18.**What is the interface available in Selenium?**

**1.Select**

**2.WebDriverWait**

**3.WebDriver //**

**4.Selenium**

**19.Select the option which is not a WebElement Method.**

**1.**getText()

**2.** clear()

**3.size() //**

**4.** sendKeys()

20. **Identify the option which is not a JAVA class in Selenium.**

**1.Select**

**2.WebDriverWait**

**3.WebDriver //**

**4.Selenium**

**21.What are the Wait applicable in Selenium webdriver.**

**1.** implictlyWait

**2.WebDriverWait**

**3.** FluentWait

**4.** All of the option //

22. **What is return type of driver.get(String url) method in Selenium.**

**1.** String

**2.** void //

**3.** null

**4.** empty space

23. **Select the Locator which is not available in Selenium WebDriver**

**1.id**

**2.tagname**

**3.name**

**4.type //**

24. **What is the javascript method used to scroll to particular Element?**

**1.** ((JavascriptExecutor) driver).executeScript( element);

**2.** ((JavascriptExecutor) driver).executeScript(“arguments[0].scrollIntoView();”, element); //

**3.** ((JavascriptExecutor) driver).executeScript(“arguments.scrollIntoView(true);”, element);

**4.** ((JavascriptExecutor) driver).executeScript(“arguments[0].scrollTo(true);”, element);

25. **What is the return Type of driver.getTitle() method in Selenium WebDriver.3**

**1.** String //

**2.** Character

**3.** void

**4.** null

* navigate().forward()
* navigate.goTo()
* navigate().to("URL")