**Tell me about yourself?**

**2 How does Page Object Model work?**

is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance

Under this model, for each web page in the application, there should be a corresponding Page Class. This Page class will identify the WebElements of that web page and also contains Page methods which perform operations on those WebElements. Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be waitForPaymentScreenDisplay().

**What framework do we use?**

\* Data Driven Framework

\* Keyword Driven Framework

\* Hybrid Framework

**What is the difference between driver.get() and driver.navigate to()?**

It does same job, one just easier to type

**What is the difference between assert and verify?**

Both of them return TRUE or FALSE. If assert returns false, it will not execute next like, however, verify will keep executing next line(s)

**How many waits in selenium?**

*Implicit* – is used to default waiting time

*Explicit* – is used halt the execution until condition is met or when time is elapsed.

**How does selenium work?**

**Advantage and disadvantage of selenium?**

**What functions has Xpath?**

**What is the polymorphism?**

*Polymorphism*. - Polymorphism contains two words "poly" and "morphs". Poly means many and Morphs means form, shape. By polymorphism, we understand that one task can be performed in different ways. For example You have a class animal, and all animals speak. But they speak differently. Here, the "speak" behavior is polymorphic in the sense and depends on the animal. So, the abstract "animal" concept does not actually "speak", but specific animals (like dogs and cats) have a concrete implementation of the action "speak".

**What is the inheritance?**

*Inheritance* - Inheritance is the most important aspect of object-oriented programming which simulates the real world concept of inheritance. It specifies that the child object acquires all the properties and behaviors of the parent object.

By using inheritance, we can create a class which uses all the properties and behavior of another class. The new class is known as a derived class or child class, and the one whose properties are acquired is known as a base class or parent class.

It provides re-usability of the code.

**What is the method overloading and method overwriting?**

**What is built-in functions in class in python do you know?**

**How do you use Property in Python and what the purpose of using it?**

**How you can select option in drop down menu?**

3 ways of selecting option in drop down:

a) by value b) by index c) by visible text

**How you can select the last option in drop down menu?**

**How you can declare private and protected variable in Python? How you call them?**

**What is the difference between them?**

**Can you overload the iframe?**

**How do you switch to the second iframe, click button A, switch to third iframe and click the button B?**

**-----------------------------------------------------------------------------**

**Questions were:**

**1. How many locators in Selenium**

\* ID (fastest)

\* ClassName

\* Name

\* TagName

\* LinkText

\* Partial Link Text

\* XPATH (slowest)

\* CSS

**2. Which of them fastest one and which of them slowest one?**

1. *IDs* are the safest, fastest locator option and should always be your first choice

2. *CSS* and *Name* selectors

Faster than XPath

3. *XPath* locators

Most flexible in order to build reliable web element locators

Very slow locator since in order to locate the element it needs to traverse the whole DOM of the page which is a time consuming operation

**3. How you count how many link you have on the page?**

**4. How you will print the text of the link?**

**5. How you count how many images on the page?**

**6. How you will go from one window to the third window?**

**7. What is the difference between find element and find elements?**

**8. How you will reversed the given string?**

**9. What is the difference between Webdriver and ChromeDriver?**

**10. How you will handle exception?**

**11. Tell me the agile process in you Company?**

**12. What is the difference between bug and defect?**

**13. What is the difference between smoke and sanity?**

**14. What is the difference between retest and regression?**

**15. Test Strategy and Test Plan is the same or not?**

**16. Tell me the folder structure in your framework?**

**18. What is the selenium version you are using?**

**19. How you find the duplicate in SQL?**

**---------------------------------------------------------------------------**

**What is the multi-thread?**

**What you would pay attention first when you are testing web-based application?**

**What will be on your prioritization list when you are testing the application?**

**Time complexity when you are writing you code? Big O**

**Do you know Selenium Suite?**

1. Selenium IDE (Integrated Development Environment)
2. Selenium RC (remote control)
3. WebDriver
4. Selenium Grid

**Do you know selenese?**

Selenese is a special test scripting language used by selenium for creating test cases. It a cross platform language used for representing the selenium commands. These are also referred to as selenese commands. Test scripts are created by writing these commands in proper sequence.

Making use of the various selenese commands a user can perform various actions such as checking for the existence of various UI elements, check for links present, enter data into text fields, select from drop down lists, etc. Besides these they can also perform actions such as switch between windows, scroll through a page, test Ajax functionality and many other web application features.

**Can you name the set of selenese commands?**

Selenese commands are of 3 types mainly:

1. *Actions* - Actions are mainly used for manipulating the state of the web application. Some of the simple actions include clicking a UI element, selecting an option form a drop-down list, typing a text into an input field, etc.
2. *Accessors* - These commands are used for checking the state of the web application and storing the result in some variables. Some of the commonly used assessor commands are ‘storeTitle’ and ‘storeTextPresent’.
3. Assertions - Assertion commands are used like checkpoints within the test script to verify that the state of the application conforms to an expected state specified by the user. Simple examples include the ‘assertText’ and ‘verifyElementPresent’.

**What is start points and end points in Selenium?**

**Locators in Selenium?**

* id,
* name,
* class\_name,
* xpath,
* link\_text,
* partial\_link\_text,
* tag\_name
* css\_selector

**Difference between locators?**

<https://blog.mozilla.org/fxtesteng/2013/09/26/writing-reliable-locators-for-selenium-and-webdriver-tests/>

**Where do you find locators?**

**What HTML is stands for? –** Hyper Text Markup Language

**How HTML is structured?**

html

head

/head

body

…

/body

/html

**What CSS stands for? –** Cascading Style Sheets

**What purpose of that?**

Define text styles, table sizes and etc

**What is POM? Tell me how it works?**

Project Object Model- Design pattern, for enhancing test maintenance and reducing code duplication.

**All concepts of OOP?**

Let us create working methods and variables, then re-use all or part of them without compromising security

1. *Object* - The object is an entity that has state and behavior. It may be any real-world object like the mouse, keyboard, chair, table, pen, etc.

Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute \_\_doc\_\_, which returns the doc string defined in the function source code.

1. *Class* - The class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class then it should contain an attribute and method, i.e. an email id, name, age, salary, etc.
2. *Method* - The method is a function that is associated with an object. In Python, a method is not unique to class instances. Any object type can have methods.
3. *Inheritance* - Inheritance is the most important aspect of object-oriented programming which simulates the real world concept of inheritance. It specifies that the child object acquires all the properties and behaviors of the parent object.

By using inheritance, we can create a class which uses all the properties and behavior of another class. The new class is known as a derived class or child class, and the one whose properties are acquired is known as a base class or parent class.

It provides re-usability of the code.

1. *Polymorphism*. - Polymorphism contains two words "poly" and "morphs". Poly means many and Morphs means form, shape. By polymorphism, we understand that one task can be performed in different ways. For example You have a class animal, and all animals speak. But they speak differently. Here, the "speak" behavior is polymorphic in the sense and depends on the animal. So, the abstract "animal" concept does not actually "speak", but specific animals (like dogs and cats) have a concrete implementation of the action "speak".
2. *Encapsulation* - Encapsulation is also an important aspect of object-oriented programming. It is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.
3. *Data Abstraction* - Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonym because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things so that the name captures the core of what a function or a whole program does.

**How .feature file is look like?**

**How many assertions do you know? Tell me how they work?**

**How many waits in Selenium? How they work?**

1. *Implicit* - An implicit wait tells WebDriver to poll the DOM for a certain amount of time when trying to find any element (or elements) not immediately available. The default setting is 0. Once set, the implicit wait is set for the life of the WebDriver object.
2. *Explicit* - An explicit wait is a code you define to wait for a certain condition to occur before proceeding further in the code. The extreme case of this is time.sleep(), which sets the condition to an exact time period to wait. There are some convenience methods provided that help you write code that will wait only as long as required.

**What is the HashMap?**

**What is the constructor?**

Constructors are generally used for instantiating an object.The task of constructors is to initialize(assign values) to the data members of the class when an object of class is created.In Python the \_\_init\_\_() method is called the constructor and is always called when an object is created.

**============================**

**Do you work with big data?**

**What main 4 functions you should pay attention while you database testing? (I didn’t get this question)**

**Components of SQL?**

**DDL?**

**DML?**

**DCL?**

**How index works in SQL?**

**What type of indexes do you know?**

**Tell me about Joins?**

**Difference between Joins and Set?**

**What is constraints? What type of constraints do you know?**

**What is difference between numbers and integers in SQL?**

**What is difference between database testing and ETL testing?**

**What is the ETL testing?**

**How ETL testing works?**

**What is the triggers in SQL?**

**What is the view in SQL?**

**What is the store procedures in SQL?**

**————————————-**

**What is the sprint velocity?**

**What is bug’s severity and priority?**

**What is team capacity?**

**Content of the Test Plan?**

**What is verification and validation?**

**————————————-**