Interview questions

* **What is inheritance** - Extending one class into other class is known as Inheritance. It can be defined as a mechanism where a new class (subclass) is derived from an existing class(superclass) properties. In Java inheritance is declared using the extends keyword. You can declare that one class extends another class by using the keyword extends in the class definition.  
  extends keyword used for Class-to-Class relationship.  
  extends keyword used for interface-to-interface relationship  
  implements keyword used with Interface to the class interface

Child Class / Sub Class : The class that extends the features of another class is known as child class, subclass, or derived class.  
*Private members of the superclass are not directly accessible to subclass, as private members' scope is only the particular class where they are declared.*

*Superclass members with default access are accessible to subclass only if they are in the same package. Default access is nothing but the state where you do not specify public, private, protected.  
  
Constructors are not inherited by subclasses.*

*We can override the method of Superclass in the Subclass, this most important when you inherit an interface.*

*If a class is Final, then we cannot inherit that class.*

There are four types of inheritance in java; those are :

1. Single Inheritance
2. Multi-level Inheritance
3. Multiple Inheritance
4. Hybrid Inheritance

Parent Class / Superclass : The class whose properties and functionalities are used(inherited) by another class is known as a parent class, superclass, or Base class.

About project – Self explanatory

**Explain Selenium framework** – To make a good tea, all the ingredients we are adding should be in a correct ratio. In case you want to make it on daily basis how you do. Its not possible to add all the ingredients in right ratio on every day. If you add all the ingredients in right ratio in a jar(to be more specific- a tea vending machine). You can make the tea every time with same taste.

Here the ‘**jar**’ where we added all the ingredients required to make a good tea is Framework.

Selenium framework’s code structure is a proportion of code re usability, readability, Reduced tests time execution, and allows multiple users to work on the same piece of the program.

**Implicit wait & Explicit wait**- When a page is loaded by the browser the elements which we want to interact with may load at different time intervals.

Not only it makes this difficult to identify the element but also if the element is not located it will throw an "**ElementNotVisibleException**" or NoSuchElement exception. Using Selenium Waits, we can resolve this problem.

The **Implicit Wait in Selenium** is used to tell the web driver to wait for a certain amount of time before it throws a "No Such Element Exception". Also it won’t wait until the set up timeout if the element is found on an early stage.

The **Explicit Wait in Selenium** is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or maximum time exceeded before throwing "ElementNotVisibleException" exception. Basically, we can use this type of wait in performance testing. Assume you have a Promocode input bar in the page source which need to show up the offer percentage within 2 secs from you enter the code and submit. But here if the price cut off element response is taking 4 secs which the implicit wait will not report if it is set to 5 secs timeout globally. But by using explicit wait we can figure out.

**How to execute TC** **simultaneously** –Basically we use Testng framework for parallel execution. TestNG provides an auto-defined XML file, where one can set the parallel attribute to method/tests/classes

<suite name="Parallel\_Testing" parallel="methods" thread-count="2">

**What is pom**-It is a design pattern which creates Object Repository for web UI elements.  
Major key advantage of POM is we will create a separate class file which would find web elements, fill them or verify them. This class can be reused in all the scripts whichever using that element. In future, if there is a change in the web element, we need to make the change in just 1 class file and not 10 different scripts.

Benefits of pom-

1. Easy to understand.
2. object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools.
3. Code becomes less and optimized because of the reusable page methods in the POM classes.

**What is page object repositories** – Data.prop

**What is agile methodology** -

Agile method proposes incremental and iterative(evolving) approach to software design

The **agile process** is broken into individual modules of a project that developers and testers work on. Every iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released. Some of the advantages in Agile are Error can be fixed in the middle of the project.

Scrum is an Agile development method that emphasizes on team-centric approach. It advocates participation of the entire team in all the project development activities.

One of them is Early Testing that focuses on −

* Writing Test Cases to express the behavior of the system.
* Early Defect Prevention, detection and removal.
* Ensuring that the right test types are run at the right time and as part of the right test level.

In Scrum, the Testing activities include −

* Contributing to User Stories based on the expected behavior of the System depicted as Test Cases
* Release Planning based on Test Effort and Defects
* Sprint Planning based on User Stories and Defects
* Sprint Execution with Continuous Testing
* Regression Testing after the completion of Sprint
* Reporting Test Results
* Automation Testing

The other commonly used Agile Testing Methodologies are −

* **Test-Driven Development (TDD)** − Test-Driven Development (TDD) is based on coding guided by tests.
* **Acceptance Test-Driven Development (ATDD)** − Acceptance Test-Driven Development (ATDD) is based on communication between the customers, developers and testers and driven by pre-defined Acceptance Criteria and Acceptance Test Cases.
* **Behavior-Driven Development (BDD)** − In Behavior-Driven Development (BDD) testing is based on the expected behaviour of the software being developed.

**What is interface**- Simple example of Interface In selenium WebDriver Is WebDriver Interface. When we are Initializing any browser using selenium WebDriver, You are writing statements like bellow.  
WebDriver driver = new ChromeDriver();

Simple example for why we use interface: Interface is created to design the structure of any project.  Suppose we create structure for drawing all the Mathematical Shapes in an interface and use this structure for various shapes like rectangle, square, cuboid. etc.

 An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class. It is used to achieve total abstraction. Since java does not support multiple inheritance in case of class, but by **using interface** it can achieve multiple inheritance.   
**interface** Shapes {

**void** draw();

**default** **void** msg(){System.out.println("default shape");}

}

**class** Rectangle **implements** Shapes{

**public** **void** draw(){System.out.println("drawing rectangle");}

}

**class** TestInterfaceDefault{

**public** **static** **void** main(String args[]){

Shapes d=**new** Rectangle();

d.draw();

d.msg();

}}

**Why we use abstract class and complete abstraction.**  
We can achieve abstraction (i.e. hiding the complex details and showing the required details) using abstract classes and interfaces. (Abstraction example - Car Class can be abstracted by showing only the car body, speed, cost, mileage and hiding the things like engine and all the machinery used).

Vehicle->Tracking system->Driving

An **interface** is like a "purely" **abstract** class. The class and **all** of its methods are **abstract**. An **abstract** class can have implemented methods but the class itself cannot be instantiated (useful for inheritance)... If you implement the **Interface** then you must implement the methods in the **interface**

Coming to the difference between abstract classes and interfaces:

First difference - Both are used for abstraction, but abstract class can have both abstract and non-abstract methods, where as interface must have only abstract methods).

Second difference - An abstract class is a class that is declared with [abstract keyword.](https://www.geeksforgeeks.org/abstract-keyword-in-java/)

while interface is a interface, means by extending abstract class you can not extend another class because Java does not support multiple inheritance but you can implement multiple inheritance via implementing interfaces in Java. i.e.

**What is annotation in testing:**   
TestNG Annotation is a piece of code which is inserted inside a program or business logic used to control the flow of execution of test methods.

Suite>Tests>Class>Method

@BeforeTest,AfterTest,BeforeSuite,AfterSuite, BeforeMethod, AfterMethod

**Advantage of Testng:**

1. TestNG provides parallel execution of test methods
2. It allows to define dependency of one test method over other method
3. It allows to assign priority to test methods
4. It allows grouping of test methods into test groups
5. It has support for parameterizing test cases using @Parameters annotation
6. It allows data driven testing using @DataProvider annotation
7. It has different assertions that helps in checking the expected and actual results
8. Detailed (HTML) reports

Interview Question:

**Tell me about yourself** - Self

**Bug life cycle**-

**Tell me about the testing process**-

**Tell me what is JIRA**- This software is used for bug tracking, issue tracking, and project management. ... The basic use of this tool is to track issue and bugs related to your software

**Tell me about trackability matrix:** The **traceability matrix** is typically a worksheet that contains the requirements with its all-possible test scenarios and cases and their current state, i.e. if they have been passed or failed. This would help the testing team to understand the level of testing activities done for the specific product.

**Tell me difference between priority and severity in JIRA**- severity = how bad it is, priority = when you are planning work, how urgently this should be looked at.

**What have you done on postman-** API testing

**What is an API:** When you use an application on your mobile phone or laptop, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions, and sends it back to our end. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API

**Types of API:** Open API, Internal.

**Locators in selenium:** CSS, XPATH, ID, CLASSNAME, LINKTEXT.

Define XPath: XPath is defined as XML path. It is a syntax or language for finding any element on the web page using the XML path expression.

Define CSS selector: the **CSS Selector** combines an element **selector** and a **selector** value that can identify particular elements on a web page.

Mandatory fields in JIRA

1.WAP to print Fibonacci series :

**public** **class** fibotest {

**static** **int** *n1*=0,*n2*=1,*n3*=0;

**static** **void** printfibo(**int** count) {

**if**(count>0) {

*n3*=*n1*+*n2*;

*n1*=*n2*;

*n2*=*n3*;

System.***out***.print(" "+*n3*);

*printfibo*(count-1);}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** count=10;

System.***out***.print(*n1*+" "+*n2*);

*printfibo*(count-2);

}

}

2.WAP to launch webdriver in eclipse

**private** **static** WebDriver *driver*;

{  
System.*setProperty*("webdriver.chrome.driver", *path*+"\\chromedriver.exe");

ChromeOptions options=**new** ChromeOptions();

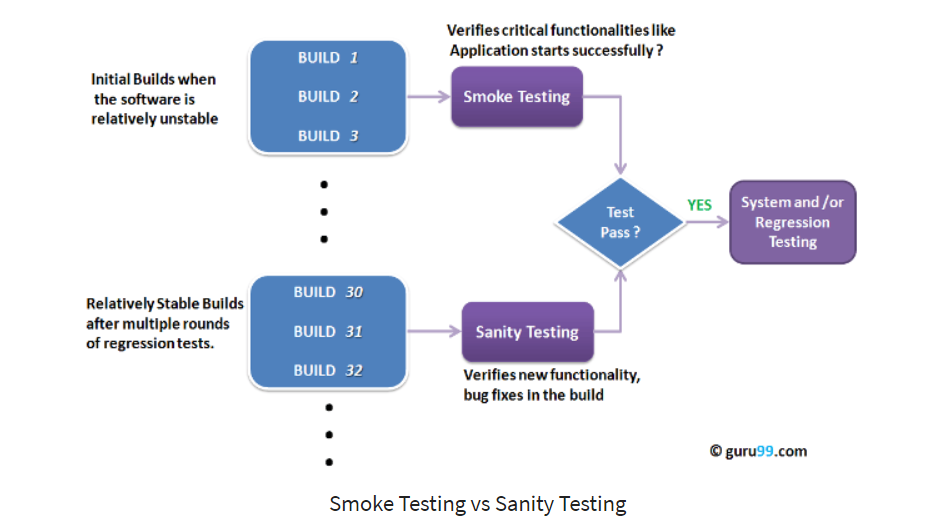
options.addArguments("--disable-notifications");

options.addArguments("start-maximized");

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

*driver*.manage().timeouts().implicitlyWait(1,TimeUnit.***SECONDS***);}

3**. OPPs concept**- <https://qascript.com/java-oops-concepts-in-selenium-automation-framework/>

4**. Sanity and smoke testing**  


|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke Testing is performed to determine whether the critical functionalities of the program works fine | Sanity Testing is done to check the new functionality/bugs have been fixed |
| The objective of this testing is to verify the "stability" of the system to proceed with more rigorous testing | The objective of the testing is to verify the "rationality" of the system to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing in software testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of [Regression Testing](https://www.guru99.com/regression-testing.html) |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

The main purpose of smoke testing is to reject a software application with defects so that QA team does not waste time testing broken software application.

**5. Regression and retesting**In the Regression Testing, the test cases are drawn out/taken out from [functional testing](http://www.indiumsoft.com/qa-services/core-qa-services/functional-testing-services/) to ensure that no new defects are included due to the software update/change & to check whether original features and functionality are working as expected. Once the regression test suite is created, the test cases can be automated using automation tool but the same is not applicable for Re-testing.  
Retesting is done to make sure that the tests cases which failed in last execution are passed after the defects are fixed.

6. How will you map your req. In ALM

**7. Diff. btw waterfall and Agile**

|  |  |
| --- | --- |
| **Agile** | **Waterfall** |
| It separates the project development lifecycle into sprints. | Software development process is divided into distinct phases. |
| It follows an incremental approach | Waterfall methodology is a sequential design process. |
| Agile methodology is known for its flexibility. | Waterfall is a structured software development methodology so most times it can be quite rigid. |
| Agile can be considered as a collection of many different projects. | Software development will be completed as one single project. |
| Agile is quite a flexible method which allows changes to be made in the project development requirements even if the initial planning has been completed. Test plan is reviewed after each sprint. | There is no scope of changing the requirements once the project development starts. Test plan is rarely discussed during the test phase. |

1. Diff. btw hashset and hashtable.

|  |  |  |
| --- | --- | --- |
| **HashMap** | **HashSet** | **Hashtable** |
| It allows null for both key and values | HashSet permits to have a single null value | It does not allow null for both key and value |
| HashMap does not maintain any order | HashSet does nto maintain any insertion order, cause insertion order is not constant overtime. But if we use LinkedHashSet it maintains an order | Hashtable does not maintain insertion order |
| HashMap uses put method to insert into hashmap | HashSet uses add method to insert into hashset | HashTable uses put method to insert into hashtable |
| HashMap is not Synchronized, better performance | HashSet is not Synchronized but can be synchronized externally | HashTable is Synchronized (thread safe) |

9. BDD and TDD framework related questions

10.Diff. btw Scenario and scenario outline

11. Sprint retro means

12. TestNG annotations

13. How will create project in Jenkins and how to integrate with Selenium.

14. How will you take pull and push.

15. Window handle and window handles diff

16. / and // difference

17. Types listeners you used in your prj.

18. Explain about Maven.

19. How will you handle popups.