1. Write a java program for the largest number from three number

Ans : **import** java.util.Scanner;

**public** **class** Largestnumber {

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

**int** x,y,z;

Scanner s= **new** Scanner(System.***in***);

System.***out***.println("Enter the first number: ");

x=s.nextInt();

System.***out***.println("Enter the second number");

y=s.nextInt();

System.***out***.println("Enter the third number");

z=s.nextInt();

**if** (x>y)

{

**if** (x>z)

{

System.***out***.println("X is greater " +x);

}

**else**

{

System.***out***.println("Z is greater " +z);

} }

**else**

{

**if** (y>z)

{

System.***out***.println("y is greater " +y);

}

**else**

{

System.***out***.println("z is greater " +z);

} } } }

1. What is SDLC and STLC? And Explain its phases.

Ans: SDLC: [Software Development Life Cycle [SDLC]](https://www.javatpoint.com/software-development-life-cycle) is a classification of individual activities executed throughout the software development process.

The SDLC includes various phases, and each phase has several activities, which help the development team design, create, and deliver a high-quality product.

The various phase included in **Software Development Life Cycle** are as follows:

* Requirements Collection
* Feasibility Study
* Design
* Programming or Coding
* Testing
* Installation
* Maintenance

STLC: [Software Testing Life Cycle [STLC]](https://www.javatpoint.com/software-testing-life-cycle) is the order of different activities executed throughout the [software testing](https://www.javatpoint.com/software-testing-tutorial) process. Testing itself has many phases called STLC, and each activity is done to improve the quality of the software product.

The various phase includes in **Software Testing Life Cycle** are as follows:

* Requirement collection or System study
* Test Plan
* Write test case
* Traceability Matrix
* Defect Tracking
* Test Execution Report
* Retrospect meeting

1. Define your roles and responsibilities.

Ans:

* Writing the test cases and validations and getting reviewed by Test Lead.
* Effective coordination between the development team and testing team.
* Bug Reporting
* Retesting and Regression testing of new features.
* Automation Testing in Selenium Web Driver with java
* Work experience in Agile methodology

1. What is regression testing?

Ans: Regression testing is a type of software testing that verifies that changes to an application's code or features don't negatively impact existing functionality. It's a key part of the software development process and is crucial for maintaining quality and reliability.

1. What are the different methodologies of SDLC? Explain each.

Ans: Software Development Life Cycle (SDLC) methodologies include:

* Agile

An iterative and incremental methodology that emphasizes collaboration and flexibility. It's best suited for small organizations, especially startups, and involves working on small portions of the project in short time frames.

* Lean

An approach that focuses on reducing waste in all stages of the project, including cost, scheduling, and scope.

* DevOps

A collaborative methodology that combines Agile and Lean, and involves close collaboration between development and operations teams.

* Spiral

A flexible methodology that combines the Waterfall and Prototyping methodologies. It's often used for large, complex projects, and involves four stages: requirements identification, design, construction, evaluation, and risk assessment.

* Iterative

An SDLC methodology that emphasizes repetition and evaluates phases until the desired results are achieved. It's similar to the Agile model, but with less scope for external involvement.

* V-model

An extension of the Waterfall SDLC approach that involves a step-by-step process after testing and coding. Early testing is typical for V-model SDLC projects.

* Big Bang model

A flexible model that doesn't follow a rigorous process or procedure, and is often used when the customer isn't sure what they want

1. Define Agile?

Ans: Agile is a project management framework and mindset that involves breaking projects into phases, or sprints, and then reflecting and adjusting the strategy after each sprint.

1. Define Scrum and Sprint?

Ans: Scrum: Scrum is a project management framework that helps teams work together to achieve a common goal. It's based on the principles of transparency, inspection, and adaptation, and it's often used by software teams.

Sprint: A sprint is **a short, time-boxed period when a scrum team works to complete a set amount of work**. Sprints are at the very heart of scrum and agile methodologies.

1. What is the estimation in Sprint?

Ans: A sprint estimation shows how much effort a series of tasks requires. It's based on assumptions, requirements, and dependencies of a project.

1. What is sprint backlog?

Ans: A sprint backlog is a list of tasks and activities that a project team plans to complete during a sprint. It's a key output of sprint planning, and it's made up of three parts:

* Sprint goal: Why the sprint is being done
* Product backlog items: What items are being selected for the sprint
* Action plan: How the sprint goal will be achieved

A sprint backlog is created by selecting items from the product backlog during a sprint planning session. The team then updates the backlog throughout the sprint as they learn more.

A sprint backlog helps prevent scope creep by clearly defining what the team will be doing during each sprint. It also helps keep the team on track to achieve the sprint's goals.

1. What are the different reports in Testing?

Ans: There are several types of test reports, including:

* Test incident report

This report documents any deviations from expected test conditions during the testing process. These incidents are reported as defects or problems and added to a repository.

* Defect report

This report lists all the defects, errors, bugs, and other discrepancies found while testing the software.

* Test cycle report

This report includes a set of test cases required for achieving specific testing goals. It also provides information on the product progress through various stages.

* Test summary report

This report provides a detailed insight into the process of testing, including the steps taken to meet the exit criteria or the specified requirements.

* Traceability matrix

This matrix maps requirements to test cases, ensuring that each requirement has been adequately tested.

* Exit criteria

This report assesses if certain conditions have been satisfied, such as whether all planned test cases were executed, critical issues were resolved, and plans for other low-priority open issues were made.

1. What are the key components of the TestCase report?

Ans: A test report should also include the following key components:

* Introduction
* Test environment
* Test execution summary
* Detailed test results
* Defect summary
* Test coverage
* Conclusion and recommendations

1. What are the components of a defect report?

Ans: A defect report should include the following components:

* Identification: A unique ID to identify the defect
* Reporter: The name and contact information of the person who reported the defect
* Application: The name and version of the application where the defect was found
* Environment: The test server or environment where the defect was found
* Browser and OS: The browser and operating system used, including version numbers
* Reproducibility: Whether the defect is reproducible, and if so, how
* Test case: A link to the test case, if applicable
* Attachments: Screenshots, videos, log files, or other relevant attachments
* Configuration settings: Any configuration settings used
* Steps to reproduce: A detailed description of the steps to reproduce the defect, including the expected and actual results
* Severity or priority: The severity or priority of the defect, based on a classification system
* Troubleshooting notes: Any troubleshooting notes

1. What is Jira?

Ans: Jira is a project management tool that helps teams plan, track, and manage their work. It is used by teams in a variety of industries, including software development, marketing, IT, design, and operations.

Here are some of the things you can do with Jira:

* Plan and track work: Jira can help you break down big ideas into achievable steps, organize work, and create milestones.
* Automate workflows: Jira can help you automate your workflows.
* Integrate with add-ons: Jira can integrate with multiple add-ons for test case management.
* Customize your workflow: Jira is flexible and can be customized to work with your team's unique workflow.
* Improve productivity: Jira can help you streamline your work processes and improve productivity.

1. How do you log defects in Jira?

Ans: To log defects in Jira, you can do the following:

1. Select Items from the top navigation bar
2. Select Defects
3. Select the Add Defect button
4. Enter the defect's name in the Defect Title field
5. Select the defect's priority, severity, status, and state
6. Select the program the defect belongs to
7. Enter values for the remaining fields as needed
8. Click Save
9. How do you link bugs with the user story?

Ans: Here are some ways to link a bug to a user story:

* In JIRA: Click on the story, then click on Link issues in the panel that opens on the right side. Search for the issue, select it, and click Link.
* In Azure DevOps: Open a bug, then find the Info panel on the right side. Click the User Story field, then select an existing story from the lookup. You can also link a bug to a user story by opening a user story, finding the Bugs tab, clicking the Magnifier icon, and finding the bug.
* In Azure DevOps, using the Board view: You can attach bugs to stories using the Board view.

You can also convert a bug into a user story by creating a subtask of the bug, associating the subtask to a user story, and then converting the subtask into a user story.

1. What is sprint?

Ans: A sprint is **a short, time-boxed period when a scrum team works to complete a set amount of work**. Sprints are at the very heart of scrum and agile methodologies.

1. Define black box and white box testing

**Black Box Testing** is the method that does not consider the internal structure, design, and product implementation to be tested. In other words, the tester does not know its internal functioning. The Black Box only evaluates the external behaviour of the system. The inputs received by the system and the outputs or responses it produces are tested.

The **White Box Test** method is the one that looks at the code and structure of the product to be tested and uses that knowledge to perform the tests. This method is used in the Unit Testing phase, although it can also occur in other stages such as [Integration Tests](https://www.practitest.com/resource-center/article/what-is-integration-testing/). For the execution of this method, the tester or the person who will use this method must have extensive knowledge of the technology used to develop the program.

1. What are String and String buffers?

**String and StringBuffer** are core Java classes used for string handling and manipulation. While the String class represents character strings as immutable objects, StringBuffer class represents mutable character strings that can be changed after creation.

**Difference Between String and Stringbuffer**

|  |  |
| --- | --- |
| **String** | **StringBuffer** |
| String is immutable, its content cannot be changed once created. | StringBuffer is mutable, its content can be changed through its append(), insert() etc methods. |
| String operations like concat() create a new String object. | StringBuffer operations are mutable and don't create new objects on each operation. |
| String is used when you don't need to modify content frequently. | StringBuffer is preferred when you need to modify the content frequently like building a String through multiple additions. |
| String can be used as a constant. | StringBuffer should not be used for constants as its content can be modified. |
| Each time a String object is modified, a separate new object is created. | StringBuffer provides better performance as it reduces creation of multiple objects. |
| String class does not have any append method. | StringBuffer provides an append() method to add content at the end. |
| Strings are stored in the pool and if a String value is already there it won't be re-created but referenced from pool instead. | StringBuffers are not interned and a new object is always created. |
| String is stored in a constant pool in memory. | StringBuffer is stored in a non-constant pool in memory. |
| String is better for immutable/constants. | StringBuffer is better for mutable/non-constant Strings. |

1. What is the retrospective meeting?

A retrospective is a meeting held after a product ships to discuss what happened during the product development and release process, with the goal of improving things in the future based on those learnings and conversations.

1. What is a collection in Java?

The **Collection in Java** is a framework that provides an architecture to store and manipulate a group of objects.

Java Collections can achieve all the operations that you perform on data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes ([ArrayList](https://www.javatpoint.com/java-arraylist), Vector, [LinkedList](https://www.javatpoint.com/java-linkedlist), [PriorityQueue](https://www.javatpoint.com/java-priorityqueue), HashSet, LinkedHashSet, TreeSet).

1. What are query parameters and path parameters in API?

In Postman, query parameters and path parameters are both ways to specify request parameters for an API call:

* Query parameters

These are appended to the end of the request URL, separated by an ampersand (&) and preceded by a question mark (?). They are listed in key-value pairs, and can be used to specify sort, pagination, or filter operations. For example, ?id=1&type=new is a query parameter.

* Path parameters

These are part of the request URL and are separated from the query parameters by a question mark (?) and from the URL by a forward slash (/). They are referenced using placeholders preceded by a colon (:). For example, /customer/:id is a path parameter.

1. What are retesting and regression testing?

Ans: **Regression testing** is a type of software testing that verifies that changes to an application's code or features don't negatively impact existing functionality. It's a key part of the software development process and is crucial for maintaining quality and reliability.

**Retesting** is the process of verifying that a specific defect or bug that was found and reported earlier has been fixed correctly.

It involves re-executing test cases that failed in the previous test cycle due to identified issues to ensure that the reported problems have been resolved.

The focus of retesting is on the specific areas or functionalities that were affected by the reported bug.

1. What are Equivalence partition and BVA?

Software Testingis a process of finding bugs and verifying whether the developed software meets its end-user requirements or not. To do testing, testers write test cases (step-by-step procedure for how to test) which take test data values as input to check the behavior of a software. In some cases, there is a pool of input values to be tested, and writing test cases for all input data values becomes laborious and error-prone. To overcome such a situation, test case design techniques are used.

This article focuses on discussing the difference between two techniques, Boundary value analysis, and Equivalence partitioning. The following topics will be discussed here:

1. **What is Boundary Value Analysis (BVA)?**
2. **What is Equivalence Partitioning?**
3. **BVA vs Equivalence Partitioning**

Let’s start discussing each of these in detail.

**What is Boundary Value Analysis (BVA)?**

[BVA](https://www.geeksforgeeks.org/boundary-value-analysis-triangle-problem/) is used to check the behavior of application using test data that exist at boundary values or in more easy words, for a range of input data values, boundary values (extreme end values) are used as input for testing. It is mostly used design technique as it is believed that software is most likely to fail at upper and lower limits of input data values.

**Example:**A software allows people of age 20 to 50 years (both 20 and 50 are inclusive) to fill a form, for which the user has to enter his age in the age field option of the software.

The boundary values are 20 (min value) and 50 (max value).

| **Invalid Value**  **(min-1)** | **Valid Value**  **(min, min+1, nominal value, max-1, max)** | **Invalid Value**  **(max+1)** |
| --- | --- | --- |
| 19 | 20, 21, 30, 49, 50 | 51 |

In the above table, one can clearly identify all valid and invalid test values (values consider during testing the system).

1. **Valid value:** Test values at which the system does not fail and function properly as per user requirement.
2. **Invalid Values:** test values that do not meet the system requirement.

**What is Equivalence Partitioning (EP)?**

It is also termed [**Equivalence Class Partitioning (ECP)**](https://www.geeksforgeeks.org/equivalence-partitioning-method/).It is a[Black Box Testing](https://www.geeksforgeeks.org/software-engineering-black-box-testing/)technique, where a range of input values are divided into equivalence data classes. In this, the tester tests a random input value from the defined interval of equivalence data classes and if the output for that input value is valid, then the whole class interval is considered valid and vice-versa.

**Example**: An application allow the user to enter the password of length 8-12 numbers (minimum 8 and maximum 12 numbers).

| **Invalid Equivalence Class** | **Valid Equivalence Class** | **Invalid Equivalence Class** |
| --- | --- | --- |
| <8 | 8-12 | >12 |

Let’s consider some password values for valid and invalid class

1. 1234 is of length 4 which is an invalid password as 4<8.
2. 567890234 is of length 9 which is a valid password as 9 lies between 8-12
3. 4536278654329 is of length 13 which is an invalid password as 13>12.

**Boundary Value Analysis vs Equivalence Partitioning**

Below are some of the differences between BVA and Equivalence Partitioning.

|  | **Boundary Value Analysis** | **Equivalence Partitioning** |
| --- | --- | --- |
| **Uses** | BVA considers the input data values from the defined boundaries. | Equivalence Partitioning examines input data values from the range of equivalence class intervals. |
| **Testing Values** | It considers min+1, min, min-1, nominal, max+1, max, and max-1 values as input test data values. | In Equivalence partitioning, valid and invalid ranges of equivalence classes are taken for testing developed applications. |
| **Bug Identification** | It identifies bugs at boundary values only. | It helps in identifying bugs in-between the partitioned equivalence data class. |
| **Application Areas** | It is a part of stress and negative testing. | It can be performed at any stage of software testing like unit testing. |
| **Usage Condition** | It is restricted to applications with close boundary values. | Correctness of Equivalence Partitioning is dependent on how correctly the tester identifies equivalence class. |

However, both Boundary Value Analysis and Equivalence Partitioning are used together as one helps in finding bugs at boundaries and another helps to determine bugs that exist between the defined range of input data values.

1. How do you handle dropdown in selenium?

The 'Select' class in Selenium WebDriver is used for selecting and deselecting options in a dropdown. The objects of Select type can be initialized by passing the dropdown webElement as a parameter to its constructor.

WebElement testDropDown = driver.findElement(By.id("testingDropdown"));

Select dropdown = **new** Select(testDropDown);

**How to select an option from the drop-down menu?**

WebDriver provides three ways to select an option from the drop-down menu.

**1. selectByIndex** - It is used to select an option based on its index, beginning with 0.

dropdown.selectByIndex(5);

**2. selectByValue** - It is used to select an option based on its 'value' attribute.

dropdown.selectByValue("Database");

**3. selectByVisibleText** - It is used to select an option based on the text over the option.

dropdown.selectByVisibleText("Database Testing");

1. List some of the test types that are supported by Selenium Define functional testing.

Different types of testing that we can achieve through Selenium are.

* Functional Testing
* Regression Testing
* Sanity Testing
* Smoke Testing
* Responsive Testing
* Cross Browser Testing
* UI testing (black box)
* Integration Testing

1. What is Agile working? Or What is the Agile methodology?

The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating.

1. What is sprint and product backlog?

A **sprint backlog** is a list of work items your team plans to complete during a project sprint. These items are usually pulled from the product backlog during the sprint planning session. A clear sprint backlog prevents scope creep by clarifying exactly what your team will be doing—and not doing—during each sprint.

A **product backlog** is a prioritized list of deliverables (such as new features) that should be implemented as part of a project or product development in Agile development.

1. What is priority and severity? Example

**Severity Definition**

**Severity**refers to the degree of impact a bug has on the software’s functionality. It helps assess how critical a bug is and determines the urgency of its resolution.

Here are some commonly recognized types of severity:

* **Critical Severity**

Bugs that result in complete system failure or make the software unusable fall under this category.

For example, if a banking application crashes when users try to transfer funds, it would be classified as CRITICAL severity.

* **Major Severity**

Bugs that cause significant issues or impair essential functionality but still allow the software to be usable fall into this category.

An example of MAJOR severity would be if an e-commerce website fails to process payments intermittently.

* **Minor Severity**

Bugs that do not significantly impact the software’s functionality but may cause inconveniences or minor glitches fall under this category.

For instance, if a social media platform displays an incorrect number of likes on a post, it would be considered a MINOR severity bug.

**Priority Definition**

**Priority** refers to the order in which bugs need to be addressed and fixed based on their importance. It helps in allocating resources and determining the sequence of bug resolution.

The common types of priorities are

* **High Priority**

Bugs that require immediate attention and resolution because they impact critical functionality or pose significant risks fall under this category.

For example, if an online ticket booking system fails to generate tickets after successful payment, it would be assigned a HIGH priority.

* **Medium Priority**

Bugs that need attention but do not severely impact the functionality or pose immediate risks fall into this category.

An example of the MEDIUM priority is if a mobile app occasionally freezes when users scroll through long lists.

* **Low Priority**

Bugs that have minimal impact on functionality and do not pose significant risks are classified as low priority.

For example, if a website’s footer alignment is slightly off, it would be considered a LOW-priority bug.

1. Define the OOP’s concept in Java.

Java defines OOP concepts as follows:

* **Abstraction.** Using simple things to represent complexity. We all know how to turn the TV on, but we don’t need to know how it works in order to enjoy it. In Java, abstraction means simple things like **objects**, **classes** and **variables** represent more complex underlying code and data. This is important because it lets you avoid repeating the same work multiple times.
* **Encapsulation.**The practice of keeping fields within a class private, then providing access to those fields via public methods. Encapsulation is a protective barrier that keeps the data and code safe within the class itself. We can then reuse objects like code components or variables without allowing open access to the data system-wide.
* **Inheritance.**A special feature of Object-Oriented Programming in Java, Inheritance lets programmers create new classes that share some of the attributes of existing classes. Using Inheritance lets us build on previous work without reinventing the wheel.
* **Polymorphism.**Allows programmers to use the same word in Java to mean different things in different contexts. One form of polymorphism is **method overloading**. That’s when the code itself implies different meanings. The other form is **method overriding**. That’s when the values of the supplied variables imply different meanings. Let’s delve a little further.

1. Give me examples of oops which you used in your framework.

**1. ABSTRACTION**

Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.

Let’s see an example of data abstraction in Selenium Automation Framework.

In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) and the methods in a Page Class. We utilize these locators in tests but we can’t see the implementation of the methods. Literally we hide the implementations of the locators from the tests.

For example, we don’t know how our phone works internally. We don’t bother about the internal mechanism but still, we can make calls.

**2. INTERFACE**

Basic statement we all know in Selenium is **WebDriver driver = new FirefoxDriver();**

WebDriver itself is an Interface. So based on the above statement **WebDriver driver = new FirefoxDriver();** we are initializing Firefox browser using Selenium WebDriver. It means we are creating a *reference variable (driver)* of the *interface (WebDriver)* and creating an *Object*. Here *WebDriver* is an *Interface* as mentioned earlier and *FirefoxDriver* is a *class*.

An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface.

**3. INHERITANCE**

The mechanism in Java by which one class acquires the properties (instance variables) and functionalities of another class is known as Inheritance.

We create a Base Class in the Automation Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class.

We extend the Base Class in other classes such as Tests and Utility Class.

Here we extend one class (Base Class like WebDriver Interface) into other class (like Tests, Utility Class) is known as Inheritance.

**4. POLYMORPHISM**

Polymorphism allows us to perform a task in multiple ways.

Combination of overloading and overriding is known as Polymorphism. We will see both overloading and overriding below.

**1. METHOD OVERLOADING**

We use **Implicit wait** in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS etc.,

**Action class** in TestNG is also an example of overloading.

**Assert class** in TestNG is also an example of overloading.

A class having multiple methods with same name but different parameters is called Method Overloading

**#2. METHOD OVERRIDING**

We use a method which was already implemented in another class by changing its parameters. To understand this you need to understand Overriding in Java.

Declaring a method in child class which is already present in the parent class is called Method Overriding. Examples are **get**and **navigate** methods of different drivers in Selenium.

**5. ENCAPSULATION**

All the classes in a framework are an example of Encapsulation. In [POM classes](https://www.softwaretestingmaterial.com/page-object-model/), we declare the data members using **@FindBy** and initialization of data members will be done using [Constructor](https://www.softwaretestingmaterial.com/java-tutorial/#constructor) to utilize those in methods.

Encapsulation is a mechanism of binding code and data (variables) together in a single unit.

**Other Selenium Automation Framework Concepts**

**1. CLASSES AND OBJECTS**

Classes are blueprints for creating objects. In a Selenium framework, you might have classes like

* Page Objects (representing web pages)
* Test Cases
* Utility Classes (for common functions)

Objects are instances of classes, allowing you to work with specific elements.

**2. WEB ELEMENT**

Web element is an interface used to identify the elements in a web page.

**3. WEBDRIVER**

WebDriver is an interface used to launch different browsers such as Firefox, Chrome, Internet Explorer, Safari etc.,

**4. FIND BY**

FindBy is an annotation used in Page Object Model design pattern to identify the elements.

**5. FIND ELEMENT**

Find Element is a method in POM to identify the elements in a web page.

**Conclusion**

By applying these OOP concepts in your Selenium automation framework, you can achieve:

* Better code organization and readability
* Improved maintainability and scalability
* Reusability of code components
* Easier debugging and testing
* More efficient collaboration among team members

1. What is TestNG?

TestNG is one of the most widely used open source testing framework used in automation testing suites.

TestNG includes all topics of a testing framework such as Features, installation, running test cases, annotations, dependent test, parallel tests, before and after annotations, etc.

* TestNG is a very important framework when you are actually developing the framework from scratch level.
* TestNG provides you full control over the test cases and the execution of the test cases. Due to this reason, TestNG is also known as a testing framework.
* Cedric Beust is the developer of a TestNG framework.
* If you want to run a test case A before that as a pre-request you need to run multiple test cases before you begin a test case A. You can set and map with the help of TestNG so that pre-request test cases run first and then only it will trigger a test case A. In such way, you can control the test cases.
* TestNG framework came after Junit, and TestNG framework adds more powerful functionality and easier to use.
* It is an open source automated TestNG framework. In TestNG, NG stands for "**Next Generation**".
* TestNG framework eliminates the limitations of the older framework by providing more powerful and flexible test cases with help of easy annotations, grouping, sequencing and parametrizing.

1. What is usability testing?

Usability testing is a method of testing the functionality of a website, app, or other digital product by observing real users as they attempt to complete tasks on it. The users are usually observed by researchers working for a business during either an in-person or, more commonly, a remote usability testing session.

1. What are the steps for reporting the defect in Jira?

* Click the Create button in the top navigation bar
* Select Bug as the issue type
* Choose the project for the defect
* Enter a summary for the defect
* Select the defect's priority, severity, status, and state
* Choose the program the defect belongs to
* Fill out the remaining fields as needed
* Click Save

1. Define the Structure of Selenium.
2. How will you handle the dropdown in selenium?
3. Different types of wait in selenium? Explain each of them.
4. Difference between hard and soft assertion?
5. Why are we using "WebDriver driver = new ChromeDrive ()"?
6. Why can't we write RemoteDriver driver = new ChromeDrive();
7. Explain the different Annotations in TestNG?
8. Define the Priority and Severity of the Bug?
9. How to maximize the screen in Selenium?
10. What are the different closure reports?
11. What is the difference between a Test Plan and a Test Strategy document?
12. Define Bug lifecycle of JIRA.
13. Various HTTP responses in API testing.
    1. Information Response -100 to 199
    2. Successful Response – 200 to 299
    3. Redirects – 300 to 399
    4. Client errors- 400 to 499
    5. Server errors-500 to 509