SELENIUM:

* introduce your self

Hi team, good mrng, thnx 4r giving me the opportunity to introduce myself. This rakesh reddy and am from hyd , I hv 3yr of experience I n testing in that I have 2 yr of manual and 1.2 yr of automation testing, coming to my education I was completed my bsc in Ou from 2010 to 2013. And am corrently working in abc company which located in hyd, and coming to my project

* Explain framework structure
* Cucu-> cucumber options, scenario, scenario outlin, report,

Features, glue, tags, plugings

* POM- @findby, page factory

@FindBy : which takes parameters specifying what to find. In Listing 10, we use the How parameter for the lookup and using as the lookup variable. How options include CSS, ID, and XPATH.

PAGEFACTRY:  is used to Initialize Elements of a Page class without having to use 'FindElement' or 'FindElements'.

* Driverfactory

A browser factory class with a basic **WebDriver** initialization logic will look something

* Xpath dynamic

//tagname[@Attribute='value']

* removing drop down values removing using xpath how u will handling
* JSX; scroll, alert, highlight, webelmnts, zoom,get title, get url

HIGHLIGHT:

JavascriptExecutor jsExecutor = (JavascriptExecutor) driver;

jsExecutor.executeScript("arguments[0].style.background='yellow'", emailTxt);

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

driver.switchTo().alert().accept();

driver.switchTo().alert().getText();

driver.switchTo().alert().sendKeys("Text");

SCROLL: JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,250)", "");

ZOOM: To set the browser to default zoom level ie., 100%

driver.findElement(By.tagName("html")).sendKeys(Keys.chord(Keys.CONTROL, "0"));

GET TITLE:

System.out.println("Page title is : " + driver.getTitle());

GET URL:

String strUrl = driver.getCurrentUrl();

      System.out.println("Current Url is:"+ strUrl);

* Action: normal click , context click (rightclick), double click, drag and drop (swithng from one frame to another), drag and dropby (sliddr)
* How to handle multiple windows

driver.getwindowhandle and driver.getwindowhandles

* how to switch betwen frames (main frame >> right frame>>> leftframe>> then main frm) ---defaultcontent
* get >> window size, position, tagname

driver.manage().window().getSize().getHeight()

driver.manage().window().getSize().getWidth()

driver.manage().window().getPosition().getX()

driver.manage().window().getPosition().getY()

**String TagName = driver.findElement(By.id("TextBox")).getTagName();**

* Capabalities: get browser verson, browser name, classname

Capabilities caps = ((RemoteWebDriver) driver).getCapabilities();

String browserName = caps.getBrowserName();

String browserVersion = caps.getVersion();

* Relocate the window to another position

driver.manage().window().setPosition(new Point(1315, 0));

* Resizing the window

 Create object of Dimensions class

     Dimension d = new Dimension(480,620);

Resize the current window to the given dimension

        driver.manage().window().setSize(d);

* implicit and explict wait

Implicit Wait time is applied to all the elements in the script. An implicit wait makes WebDriver poll the DOM for a certain amount of time when trying to locate an element.

Explicit Wait time is applied only to those elements which are intended by us. An explicit wait makes WebDriver wait for a certain condition to occur before proceeding further with execution

* Screenshot
* Assertions (Soft and hard); verify
* open a link in new tab and new window

driver.get("www.Url1.com");

driver.switchTo().newWindow(WindowType.TAB);

driver.get("www.Url2.com");

* navigate, close, quit,
* Group, Parameterisation (Data provider) – testing

Parameters Annotation in TestNG is **a method used to pass values to the test methods as arguments using .** **xml file**. Users may be required to pass the values to the test methods during run time. The @Parameters annotation method can be used in any method having @Test, @Before, @After or @Factory annotation.

Groups: TestNG Groups **allow you to perform groupings of different test methods**. Grouping of test methods is required when you want to access the test methods of different classes. Not only you can declare the methods within a specified group, you can also declare another group within a specified group.

* parraell execution (concurrancy)

parallel execution **enables organizations to simultaneously run tests in parallel on different devices and browsers to significantly shrink testing times**. Parallel testing is a test automation method wherein test cases are simultaneously run on multiple combinations of browsers, operating systems, and devices.

* im hvng 10 checkboxes in a page; here i need to selct all checkbox, later nly last 5 checkboxes
* Git hub; how and wht is git; push, pull, commit, branch, clone
* jhenkins; CI /CD
* selenium version, grid, framworks, maven, testng, junit, pom, jdk, jvm, jar

selenium version: 6

Grid: **it easy to run tests in parallel on multiple machines**. This is done by routing commands to remote web browser instances, where one server acts as the hub. This hub routes test commands that are in JSON format to multiple registered Grid nodes.

Framework: The **Selenium Framework** is a code structure that makes code maintenance easy and efficient. Without frameworks, users may place the “code” and “data” at the same location which is neither reusable nor readable. Frameworks produce beneficial outcomes like increased code reusability, higher portability, reduced cost of script maintenance, better code readability, etc.

Maven: is **the latest build testing tool**. It has several new features as compare to Ant, like dependency, etc. Maven is a project build or project management tool. It is used to check the compilation issues between framework components whenever multiple test engineer integrates their files into the same framework.

TestNG: is **an open-source testing framework where NG stands for 'Next Generation.** **'** It is architected to simplify a broad range of testing needs starting from unit testing to integrated system testing. Initially, both JUnit and TestNG were designed solely for unit testing

Junit: is **an open source unit testing tool and used to test small/large units of code**. To run the JUnit test you don't have to create a class object or define the main method. JUnit provides assertion library which is used to evaluate the test result. Annotations of JUnit are used to run the test method.

POM: **Page Object Model**, also known as POM, is a design pattern in Selenium that creates an object repository for storing all web elements. It is useful in reducing code duplication and improves test case maintenance.

JDK: JDK stands for **java development kit**. Install any version that is 1.7 or higher.It is advisable to have JDK and NOT JRE.

JVM: is stands for Java Virtual Machine (JVM) is **the runtime engine of the Java Platform, which allows any program written in Java or other language compiled into Java bytecode to run on any computer that has a native JVM**.

Jar: is **a group of API's rolled into one jar for different languages** (Java, Python, C#, Javascript, etc.). The client jar can be acquired by tools like Maven or Gradle, basically open-source build automation systems.

* How to read data excel sheet and how u will do this and whn u will do? what dependencies... apache poi
* whn u will do AT, and what are all the tc u will take for AT?
* invocation count in testng , priority, dependonmethods, groups

TestNG supports multi-invocation of a test method, i.e., **a @Test method can be invoked multiple times sequentially or in parallel**. If we want to run single @Test 10 times at a single thread, then invocationCount can be used. In this example, the @Test method will execute for 10 times each on a single thread.

Priority: In TestNG, Priority is **an attribute that helps the users define the order in which they want the test cases to be executed**. When you have multiple test cases and want them to run in a particular order, you can use the Priority attribute to set test priority in TestNG

Groups: TestNG Groups **allow you to perform groupings of different test methods**. Grouping of test methods is required when you want to access the test methods of different classes. Not only you can declare the methods within a specified group, you can also declare another group within a specified group.

DependsOnMethod: dependsOnMethods attribute on a test method [test1 e.g.] **specifies all the test methods [test2, test3,..]** **this test method depends on**. It means test1 will start execution only after all the tests it depends on executed successfully.

* selenium methods(wait, navigate, browser, switch, webelemnts)

driver.navigate().refresh();

driver.navigate().forword();

driver.navigate().back();

driver.navigate().to();

impicitwait()

explicitwait()

switchTo();

* netwrk latency: is sometimes called lag, is **the term used to describe delays in communication over a network**.
* whether testng has main method?

**TestNG class files don't require main method** because TestNG uses annotations like @Test to take care of the executions while running the TestNG class.

* What is a final keyword

The **final** is a **keyword in java**. **Final** can be variable, method, class or parameter.

In the Java programming language, the final keyword is used in several contexts to define an entity that can only be assigned once. Once a final variable has been assigned, it always contains the same value.

==============================================================================

Manual:

* sdlc, stlc, wht process/methodolgy (agile)

SDLC: is a software delepment life cycle

STLC: is software testing life cycle

Methodology is agile and its process is The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of **planning, executing, and evaluating**.

* how will u log a bug

Try to reproduce the bug to make sure that it is indeed a bug and not a user or environment error.

Check if the bug has already been reported.

Report the bug

#1) Bug Number/id.

#2) Bug Title.

#3) Priority.

#4) Platform/Environment.

#5) Description.

#6) Steps to Reproduce.

#7) Expected and Actual Result.

#8) Screenshot.

what is ur daily role

* smoke and sanity

Smoke Testing has a goal to verify “stability” whereas Sanity Testing has a goal to verify “rationality”.

Smoke Testing is done by both developers or testers whereas Sanity Testing is done by testers.

Smoke Testing verifies the critical functionalities of the system whereas Sanity Testing verifies the new functionality like bug fixes.

Smoke testing is a subset of acceptance testing whereas Sanity testing is a subset of Regression Testing.

Smoke testing is documented or scripted whereas Sanity testing isn’t.

Smoke testing verifies the entire system from end to end whereas Sanity Testing verifies only a particular component.

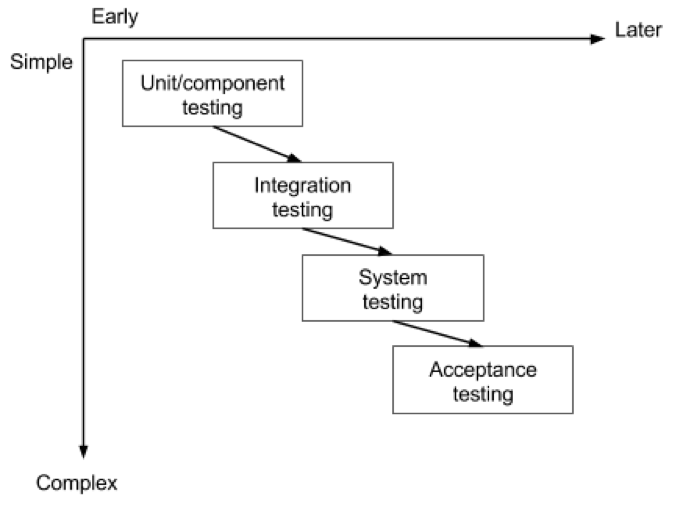
* UAT

**User Acceptance Testing (UAT)** is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

The main **Purpose of UAT** is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

* types and levls of testing

There are generally four recognized levels of testing: **unit/component testing, integration testing, system testing, and acceptance testing**.



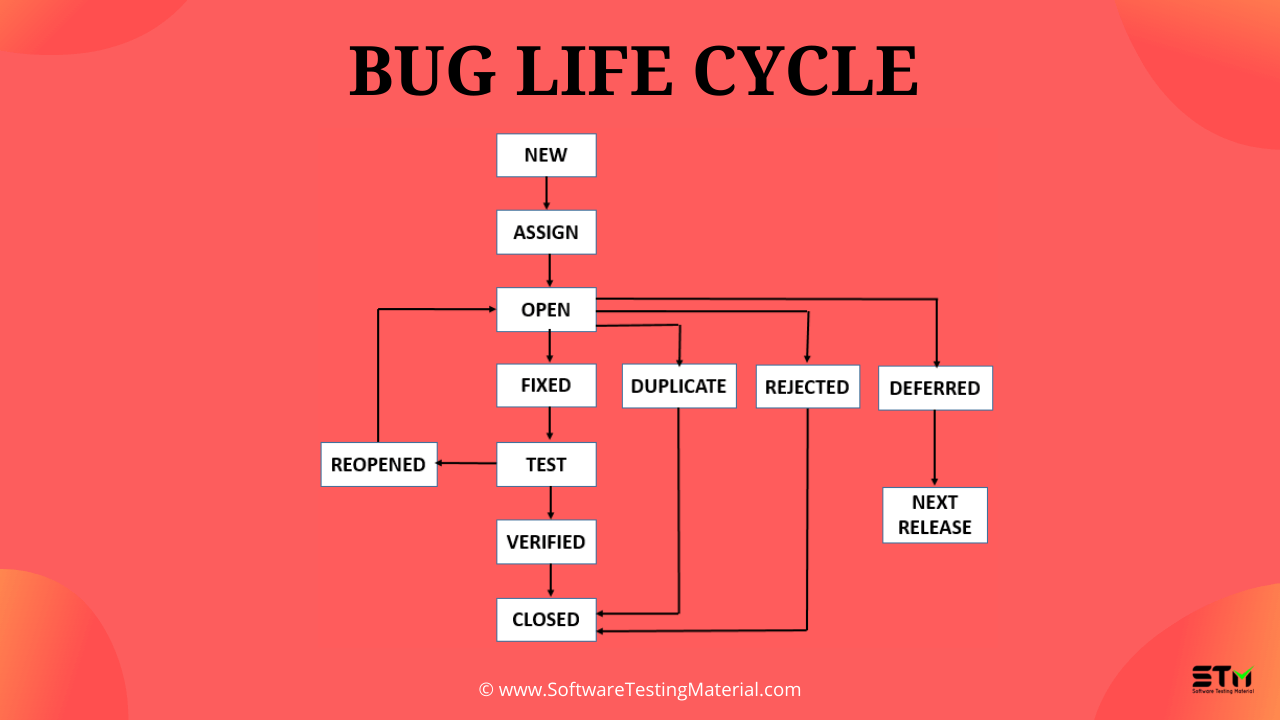
* Performance and security testing, compatability testing

Performance:  is **the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload**. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

security testing: is a type of Software Testing that **uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders**. It ensures that the software system and application are free from any threats or risks that can cause a loss.

compatibility testing: is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.

* defect life cycle



* defect relase

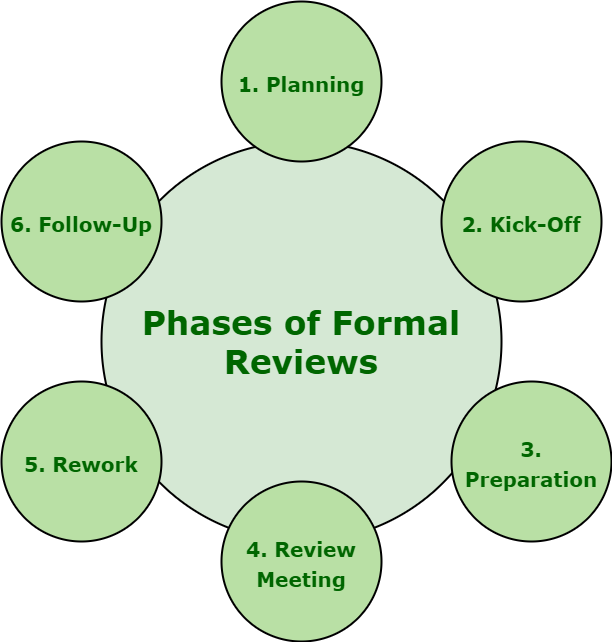
 is **when software or an application is handed over to the testing team knowing that the defect is present in a release**. During this the priority and severity of bug is low,

* agile vs traditional

| Traditional Software Development | Agile Software Development |
| --- | --- |
| It is used to develop the simple software. | It is used to develop the complicated software. |
| In this methodology, testing is done once the development phase is totally completed. | In this methodology, testing and development processes are performed concurrently. |
| It provides less security. | It provides high security. |
| It provides less functionality in the software. | It provides all the functionality needed by the users. |
| It is basically used by fresher’s. | It is used by professionals. |
| Development cost is less using this methodology. | Development cost is high using this methodology. |
| It majorly consists of five phases. | It consists only three phases. |
| It is less used by software development firms. | It is normally used by software development firms. |

* formal review

generally takes place in piecemeal approach that consists of six different steps that are essential. Formal review generally obeys formal process. It is also one of the most important and essential techniques required in static testing.



* agile (sprint , backlog, scrum, who s a scrum master) , disadvantg of agile

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, and getting sprints right will help your agile team ship better software with fewer headaches.

Scrum:  is **a framework for project management that emphasizes teamwork, accountability and iterative progress toward a well-defined goal**

Scrum Master: is a professional who **leads a team through a project using Agile project management techniques**. A Scrum master facilitates all the communication and collaboration between leadership and team players to ensure a successful outcome.

Backlog:  is **a prioritized list of deliverables (such as new features) that should be implemented as part of a project or product development**. It's a decision-making artifact that helps you estimate, refine, and prioritize everything you might sometime in the future want to complete.

* test coverage, defect density, test efficiency, defact leakage

Test Coverage: is defined as a metric in **Software Testing** that measures the amount of testing performed by a set of **test**.

Defect Density: can be defined as **the number of confirmed bugs in a software application or module during the period of development, divided by the size of the software**.

Test Efficiency: determines the efficiency of test processes. It checks the number of resources required and is actually utilized in the project. It is the number of test cases executed /unit of time. Time is generally in hrs.

Defect leakage: how many defects are missed/slipped during the QA testing. Defect Leakage = (No. of Defects found in UAT / No. of Defects found in QA testing.)

* inspection, walkthrough,

Inspection: is **a formal type of review that involves checking the documents thoroughly before a meeting and is carried out mostly by moderators**.

Walkthrough:  in software testing is **used to review documents with peers, managers, and fellow team members who are guided by the author of the document to gather feedback and reach a consensus**.

* srs, frs,

SRS: A software requirements specification (SRS) is **a document that describes what the software will do and how it will be expected to perform**. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

FRS:  functional requirement specification is **the document that describes all the functions that software or product has to perform.** it's a step-by-step sequence of all operations required to develop a product from very start to end.

* test plan, test startergy

Test Plan: refers to **a detailed document that catalogs the test strategy, objectives, schedule, estimations, deadlines, and the resources required for completing that particular project**. Think of it as a blueprint for running the tests needed to ensure the software is working properly – controlled by test managers.

Test Stratergy:  is **an outline that describes the testing approach of the software development cycle**. The purpose of a test strategy is to provide a rational deduction from organizational, high-level objectives to actual test activities to meet those objectives from a quality assurance perspective.

**Java**

* what is java, class, object, method, package, project,

JAVA: is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

CLASS: is **a template used to create objects and to define object data types and methods**. Classes are categories, and objects are items within each category. All class objects should have the basic class properties

OBJECT: A Java object is **a member (also called an instance) of a Java class**. Each object has an identity, a behavior and a state. The state of an object is stored in fields (variables)

METHOD: A method is **a block of code which only runs when it is called**. You can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as functions.

PACKAGE:  is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

PROJECT: A project is **a group of source files and the settings with which you build, run, and debug those source files**. In the IDE, all Java development has to take place within a project. For applications that involve large code bases, it is often advantageous to split your application source code into several projects.

* oops conct (Abstarction, polymorphism, inheritance, encapsulation)

OOPs: OOPs stands for Object-oriented programming. OOPs in Java organizes a program around the various objects and well-defined interfaces. The OOPs Concepts in Java are abstraction, encapsulation, inheritance, and polymorphism. These concepts aim to implement real-world entities in programs.

Abstraction: Abstraction is **the concept of object-oriented programming that "shows" only essential attributes and "hides" unnecessary information**. The main purpose of abstraction is hiding the unnecessary details from the users.

Polymorphism: is one of the core concepts of object-oriented programming (OOP) and **describes situations in which something occurs in several different forms**. it describes the concept that you can access objects of different types through the same interface.

Inheritance: is **the procedure in which one class inherits the attributes and methods of another class**. The class whose properties and methods are inherited is known as the Parent class.

Encapsulation: refers to **the bundling of data, along with the methods that operate on that data, into a single unit**. Many programming languages use encapsulation frequently in the form of classes.

* "This" and "Super" keywrd

THIS: The most common use of the this keyword is to eliminate the confusion between class attributes and parameters with the same name

SUPER: It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

* Interface ? how u will achive this

**Interface methods do not have a body - the body is provided by the "implement" class.** The interface in Java is **a mechanism to achieve abstraction**. There can be only abstract methods in the Java interface, not the method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have abstract methods and variables.

* constructor; what, whether it hv return type, whether it cn be overloaded

Constructor: A constructor in Java is **a special method that is used to initialize objects**. The constructor is called when an object of a class is created.

**Every method in Java is declared with a return type and it is mandatory for all java methods.**

**Yes, we can overload the main method in Java**, but When we execute the class JVM starts execution with public static void main(String[] args) method.

**OverRiding: No, because the main is a static method**

* how to use main method.

Java main method is the entry point of any java program. Its syntax is always **public static void main(String[] args)** .

* how to call statci and non static method

Static: ClassName.MethodName();

Non-Static: create object for class and Obj.MethodName();

* How to use methods with parameter / argumnets and return type

Return type: void – The method returns nothing. We use “void” keyword if we want a method not to return anything but perform operations only / Execute group of statements. ...

Return type: int – The method returns int data type. ...

Return type: String – The method returns String data type of value.

* Exception handling(Try, catch, finally)

The **try-catch** is the simplest method of handling exceptions. Put the code you want to run in the try block, and any Java exceptions that the code throws are caught by one or more catch blocks. This method will catch any type of Java exceptions that get thrown. This is the simplest mechanism for handling exceptions.

FINALLY: The finally block in java is used to put important codes such as clean up code. The finally block **executes whether exception rise or not and whether exception handled or not**. A finally contains all the crucial statements regardless of the exception occurs or not.

* Throw and Throws

Both throw and throws are **concepts of exception handling in Java**. The throws keyword is used to declare which exceptions can be thrown from a method, while the throw keyword is used to explicitly throw an exception within a method or block of code.

* Math class

The Math class **contains methods for finding the maximum or minimum of two values, rounding values, logarithmic functions, square root, and trigonometric functions**

**Math class** provides several methods to work on **math** calculations like min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs()

* Built in string concept (lower case to uppercase, get particular value, substring...)

Lower to upper case:

You can use **toUpperCase() to convert any lower case String to uppercase and toLowerCase() to convert any uppercase String to lowercase**.

Key thing to remember while using toUpperCase() and toLowerCase() is that they return a different string rather than modifying the same String because String is immutable in Java.

SubString:

Substring in Java is **a commonly used method of java.** **lang.** **String class that is used to create smaller strings from the bigger one**. As strings are immutable in Java, the original string remains as it is, and the method returns a new string

* Scanner class

The Scanner class is **used to get user input**, and it is found in the java.util package. To use the Scanner class

* Array

An array is **a container object that holds a fixed number of values of a single type**. The length of an array is established when the array is created. After creation, its length is fixed.

* While, do while, for loop, foe each loop

While: The Java while loop is **used to iterate a part of the program repeatedly until the specified Boolean condition is true**. As soon as the Boolean condition becomes false, the loop automatically stops. The while loop is considered as a repeating if statement. **while(true){** **//code to be executed**.

DoWhile: The do/while loop is **a variant of the while loop**. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

**do { // statements } while (expression);** The expression for the do-while loop must return a boolean value, otherwise, it will throw compile-time error.

**FOR:** **an entry-controlled loop that facilitates a user to execute a block of a statement(s) iteratively for a fixed number of times**.

FOR-EACH: the for-each loop is **used to iterate through elements of arrays and collections** . It is also known as the enhanced for loop

* Boxing and unboxing

**Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes**. For example, converting an int to an Integer, a double to a Double, and so on. If the conversion goes the other way, this is called unboxing.

* datatypes, operators

primitive , non primitive data types

Arithmetic Operators.

Assignment Operators.

Relational Operators.

Logical Operators.

* collection framework (Set, list, map,queue)
* whether set cn be converted to list

we can convert set to list by using ArrayList or linkedList constructor

* and can array be converted to list

Using Arrays. asList() Method.