Stage 1 - Core Java & advance java

Stage 2 - Selenium (4 hours (web automation) & AutoIT (windows automation)

Stage 3 - BDD framework

Stage 4 - Git and Jenkins (CI/CD pipeline)

Github

Java - <https://github.com/balaji-githubstore/JavaConceptHealthAsyst.git>

Selenium - <https://github.com/balaji-githubstore/SeleniumConceptHealthAsyst.git>

BDD Framework

Framework - <https://github.com/balaji-githubstore/OpenEMRBDDHealthAsyst.git>

Please let me know incase of any issues - balajidinakaran1@gmail.com

1. Java

* Language developer by Sun microsystems
* High level programming
* Platform independent
  + Write once run anywhere

Java Architecture

Source code (.java) 🡪 Byte code (.class) 🡪 O/P

1. JVM - Java Virtual Machine

* Source code - can be written in simple text file (.java)
* Java compiler - converting source code (.java) to byte code (.class)
* Byte code interpreted by JVM
* JVM varies from platform to platform or JVM is platform dependent

1. Installation
   1. JDK 8 (Java Development Kit)
      * By default JRE (Java Runtime Environment) will be installed

<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>

* 1. IDE - Eclipse/Intellij/netbean

Option 1-

* Download eclipse.exe
* Choose - Eclipse IDE for Java developer
* Install it

Option 2-

* Download installed package as zip folder.

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2021-12/R/eclipse-java-2021-12-R-win32-x86_64.zip>

1. Compile time - Source code to byte code

Run time - Bytecode to O/P

1. UpperCamelCase - MyFirstProject

lowerCamelCase - myFirstProject

1. Eclipse Structure

Workspace - lowercase

Project 1 - UpperCamelCase

Package 1 - lowercase (com.healthasyst.purpose) or (org. healthasyst.purpose)

Class 1 - UpperCamelCase

Methods/variables - lowerCamelCase

Class 2

Methods/variables

Package 2

Project 2

1. Datatypes
   1. Primitive datatype/pre-defined datatype
      * Byte - 8 bits

1 bit - sign (+ or -)

7 bit - storing the number

* 1. Non-Primitive datatypes/ Non - predefined datatypes - collection of primitive datatypes
* String
* Array
  + Size should be defined
  + Zero based index

1. Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 6

at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:48)

at java.base/java.lang.String.charAt(String.java:712)

at com.healthasyst.datatypes.DatatypesDemo.main(DatatypesDemo.java:44)

1. Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

at com.healthasyst.datatypes.ArrayDemo.main(ArrayDemo.java:14)

1. Debugging

* Resume
* Terminate
* Step into
* Step over

1. Conditional statement
   1. If condition
   2. Switch condition
2. Relational operators

==, !=, >,<,>=,<=

1. Logical operator

And (&&), Or (||), not

1. Iterative statements
   1. For loop
      * Initial condition
      * End condition
      * Iteration
   2. For each loop - designed for handling the array and collections
   3. While
   4. Do while - minimum once it will execute
2. break - to stop the further iteration

Continue - just moves to next iteration

1. Pre/post increment
2. **Methods - It’s a building block of your program**

* Reusability
* Maintenance
  1. **Create and call static method**

//accessmodifier static returntype methodname(arguments)

**public** **static** **double** areaOfCircle(**int** r)

{

**double** area = 3.14 \* r \* r;

**return** area;

}

**To call the static method** - **classname.methodname(args)**

* 1. **Create and call non-static method**

//accessmodifier returntype methodname(arguments)

**public** **double** areaOfTriangle(**double** base, **double** height) {

**return** (base \* height) / 2;

}

**To call the non-static method** -

* **Create object**
* **Use *objref.methodname()***

1. **Object** 
   1. **Declaration -** Area obj
   2. **Instantiation - new**
   3. **Initialize -** Area()
2. **Variable** 
   1. **Create and call static variable**
   2. **Create and call non-static variable**

1. Static vs Non-static with variable
2. Class & Object
3. Access modifier
   1. Private - accessible within the class
   2. Default - accessible within the package
   3. Protected - - accessible within the package and also in the inherited class
   4. Public - anywhere
4. SOLID Principle

S-Single Responsibility Principle - A class should have one and only reason to change, it should have only one job.

1. Exception in thread "main" java.lang.NullPointerException: Cannot read field "aNS" because "obj2" is null

at com.healthasyst.variable.VariableDemo.main(VariableDemo.java:32)

1. **this** keyword can be used under any method to refer the current object.
2. Stack and heap
3. String - immutable
4. Git - Git is a free and open source **distributed version control system.**

Download and install - <https://git-scm.com/>

**Architecture**

Project (local system) 🡪 local repository (local system) 🡪 remote repository (github, AWS code commit, Bit Bucket)

Steps

Local repo

1. git init 🡪 initialize the local git repo
2. git add README.md 🡪 collect the files,folder ready to move
3. git commit -m "first commit" 🡪 update the local repo

Server repo/remote repo

1. git remote add origin <https://github.com/balaji-githubstore/JavaConceptHealthAsyst.git> 🡪 register the remote url with the name origin
2. git push -u origin main
3. Constructor

* Pre-requisite for the object
* Constructor name and class name should be same. There is no return type.
* There will be default constructor which helps to load all the non-static (instance) variable with default value.
* We can override the default constructor by creating the constructor explicitly
  + Constructor without arguments
  + Constructor with arguments
* Whenever constructor created then we need to call that constructor on creating the object.

1. Static Polymorphism/ Compile time polymorphism/ early binding - Method to be called is resolved during compile time.
2. Constructor Overloading

By change in

* 1. Number of parameters/arguments
  2. Sequence of parameters
  3. Datatypes of parameters

1. Method Overloading

Multiple method with same name

By change in

* 1. Number of parameters/arguments
  2. Datatypes of parameters
  3. Sequence of parameters

1. Inheritance - reusability
   1. When “is-a” relationship existing between two classed, we use inheritance
   2. If a parent class is having constructor with parameter then you need to call it using super keyword.
2. Understanding more about String
   1. String - immutable
   2. String pool
   3. String builder
3. == and String method (.equals)
   1. == 🡪 compares the location and text
   2. .equals 🡪 compares only the text
4. Dynamic Polymorphism/ Method Overriding/ Run time polymorphism/ Late binding

Method to be called is resolved during run time.

* You need create the method in parent and override the same method in child with different definition. Signature, parameter should be same as parent.
* Create an object for child and store it in parent reference.
* Based on the object created, if the method is overridden then that method will be called during runtime.

1. Abstraction
   1. Abstract class
      * Can have method without definition
      * When the method is abstract then the class should be abstract
      * Whichever the class inherit the abstract class then the child class should give the definition
      * Cannot create an object for abstract class (cannot instantiate)
      * When you have method with definition (non-abstract methods )and without definition then you can go with abstract.(abstract method)
   2. Interface - to establish the set of rules/define standard

Few rules

* + - Method declared inside interface is **public and abstract method**

1. All lower level module will be depends on higher level module (like abstract class or interface)
2. Exception in thread "main" java.lang.ClassCastException: class com.healthasyst.oops.ChromeDriver cannot be cast to class com.healthasyst.oops.Employee (com.healthasyst.oops.ChromeDriver and com.healthasyst.oops.Employee are in unnamed module of loader 'app')

at com.healthasyst.oops.BrowserTest.main(BrowserTest.java:101)

1. Type Casting
2. Collections
   1. Non-Generic type
   2. Generic type (recommended)

Selenium - A suite of tools

1. Selenium IDE
   1. No need of programming knowledge
   2. Record and playback feature
   3. Plugin - chrome, firefox, edge
   4. Used for Simple scripting
2. Selenium RC (Remote Control)
   1. Programming knowledge is must
   2. Java, C#, python, ruby, javascript
   3. Architecture

Source code (java+selenium RC) 🡪 RC server (turn on/off) 🡪 Browser

1. Selenium WebDriver
   1. Programming knowledge is must
   2. Java, C#, python, ruby, javascript
   3. Architecture

Source code (java+selenium webdriver) 🡪 Browser

1. Selenium Grid
   * + Hub and Node

Selenium WebDriver

1. Create Project
2. Configure the selenium jar
3. Configure the driver based on the browser and browser version.
4. Exception in thread "main" java.lang.IllegalStateException: The path to the driver executable The path to the driver executable must be set by the webdriver.chrome.driver system property; for more information, see https://github.com/SeleniumHQ/selenium/wiki/ChromeDriver. The latest version can be downloaded from <https://chromedriver.storage.googleapis.com/index.html>

All drivers can be found at

<https://www.selenium.dev/downloads/>

To fix:

* Download the driver and keep it in project home directory
* Download the driver and use System.SetProperty to driver directory
* Download and add the driver path to environment path variable
* Configure using WebDriverManager - which will check the current browser, browser version, platform and then auto download the driver.
* Selenium webdriver doc - <https://www.javadoc.io/doc/org.seleniumhq.selenium/selenium-api/latest/org/openqa/selenium/WebDriver.html>
* Inspect - tagname, attribute, text or not
* Click, sendkeys, select
* Basic Locators
  1. Id
  2. Name
  3. Classname
  4. Tagname
  5. Linktext
  6. Partial linktext

When there is duplicate locator then findelement will pick the first one.

* Advance locator
  1. Xpath
  2. CSS
* To get By object 🡪

By.*id*("authUser")

* To get WebElement object 🡪

WebElement ele= driver.findElement(By.*id*("authUser"));

ele🡪L1-> (

<input type="text" class="form-control" id="authUser" name="authUser" placeholder="Username:">

)

* Page load 🡪 // wait for the page load to complete
* findElement 🡪 check for the presence of element in 0.5s
* Synchronization
  1. Unconditional wait

Thread.Sleep(5000) 🡪 wait for 5s

* 1. Conditional wait
     1. Implicit wait
        1. Default implicit wait - 0s
        2. Applicable for all findelement and findelements method.
        3. Example: Implicit wait = 30s
           1. If element is not present, it will check for 30s then it will throw error
           2. If element is present then it will do operation immediately
           3. Polling time 🡪 0.5s (how frequently it checks)
     2. Explicit wait

Exact condition

polling time - 0.5s

* + 1. Fluent wait
* Dropdown
  1. Dropdown with select tag
     1. selectByVisibleText(string)
     2. selectByValue(string)
     3. selectByIndex(string)
  2. Dropdown without select tag
     1. Use click operation to complete the task
* Exception in thread "main" org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"css selector","selector":"#authUser12"}
* gettext, getattribute, gettagname
* Frame, Multiple tabs/windows, alert - switchTo()
* Frame
  1. Even though the locator is correct, we will get NoSuchElementException then we need to check for frame
  2. Check for tagname - frame, iframe
  3. To get into frame
     1. Using Index
     2. Using name or id as a String
     3. Using webelement

driver.switchTo().frame(driver.findElement(By.*xpath*("//frame[@name='login\_page']")));

* List<String> vs Set<String>
  1. List can contains duplicates
  2. Set cannot contains duplicates
* Close vs quit
  1. Close - close the current tab/current session
  2. Quit - close the browser and also kill the process associated with it.
* Multiple tabs/windows
* Exception in thread "main" org.openqa.selenium.NoSuchWindowException: no such window: target window already closed
* CSS selector - https://www.w3schools.com/cssref/css\_selectors.asp
* Action - mouse/keyboard
* findelements
  1. check size to make sure element is present
* Javascript
  1. to click on hidden click
  2. on typing validation may not happen
  3. scroll
  4. scroll to element
  5. getting tooltips
  6. to avoid mouse activities

javascript - <https://www.w3schools.com/js/js_htmldom.asp>

click

document.querySelector("[type='submit']").click()

document.querySelector("li[data-bind='click: logout']").click()

type

document.querySelector('#authUser').value='balaji'

scrollIntoView

document.querySelector("#SubscriptionAgreement").scrollIntoView()

javascript & webelement

click

js.executeScript("arguments[0].click()", driver.findElement(By.*xpath*("//li[text()='Logout']")));

type

js.executeScript("arguments[0].value=’balaji", driver.findElement(By.*xpath*("//\*[@id='authUser]")));

BDD Framework: -

BDD - Behaviour Driven Development

Where to apply BDD?

1. Everyone knows about the product
2. Some of the team members knows about the product
3. Only Management team knows about the product
4. Someone/somewhere already delivered this product but in your organization it’s new
5. We need to do R&D and then need to develop

ATDD - Acceptance Test Driven Development

Modified BDD

Advantage

1. Living Documentation
2. New developer, QA, Stakeholders -
3. Better communication between entire team and product owners

Architecture of Cucumber

Feature (.feature) → Step Definition (.java)

Maven - Build Management Tool

* + - 1. Create Maven Project
         1. Group Id = com.companyname.teamname
         2. Artifact id = project name

POM.xml - Project Object Model

* + - 1. Configure the jars related to Selenium, Cucumber (BDD)

Cucumber

<https://mvnrepository.com/artifact/io.cucumber/cucumber-java/7.2.3>

<https://mvnrepository.com/artifact/io.cucumber/cucumber-junit/7.2.3>

* + - 1. Add Cucumber plugin into eclipse
      2. Create a feature file
         1. feature header
         2. feature description

Unfomatted description - free text

Formatted description

In order to [business goal]

As a [role]

I would like [visible change in the software]

* + - 1. Scenario - title of the scenario

Given [prerequisite for the current scenario]

When [actions/operation]

Then [verification]

* + - 1. Create step definition file and add the steps details
      2. Configure the driver using webdrivermanager

<https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager/5.0.3>

* + - 1. Assertion is mandatory
      2. Step Parameterization
      3. Scenario Outline - Multiple data
      4. dry = true (checks all steps having matching step definitions or not)
      5. tags - <https://cucumber.io/docs/cucumber/api/#tags>
      6. background - repeated given - only when all scenario uses same given.
      7. datatable - tabular data for one step
      8. Report generation - cucumber
         1. pretty
         2. html
      9. How to run failed scenario?
         1. Add rerun

plugin = {"html:target/index.html","rerun:target/rerun.txt"}

* + - * 1. Failed scenario detail will be added to rerun.txt file
        2. Now create runner.java file and use rerun.txt as feature argument

@CucumberOptions(

features = {"@target/rerun.txt"}

,glue = {"com.healthasyst.stepdefn","com.healthasyst.base"}

,publish = **true**

,monochrome = **true**

,plugin = {"pretty","html:target/index.html","rerun:target/rerun.txt"}

)

@RunWith(Cucumber.**class**)

**public** **class** FailedScenarioTest {

}

Improve the readability

Step parameters - to reduce duplicates in step definitions

Scenario outline - to reduce duplicates in feature file/ multiple test data

background - for repeated given

Datatable - Send bulk data/tabular data in one step

* + - 1. Page Object Model

Reusability

Maintenance

* + - * 1. For each page in the application, we need to create a class - PageClass
        2. Operation will happens through Methods - Page Method
        3. Collecting the repository of web ui elements - Page Object Repository

Page Class

LoginPage

enterUsername(string)

enterPassword(string)

selectLanguage(string)

clickOnLogin()

getLoginPageTitle()

getApplicationDescription()

clickOnAckLicCerti()

getInvalidErrorMessage()

MainPage - all menu can be handled here

clickOnCalendar()

clickOnMessages()

clickOnPatient/Client()

clickOnPatients()

getMainPageTitle()

PatientFinderPage

clickOnAddNewPatient()

SearchOrAddPatientPage

enterFirstName(string)

enterLastName(string)

…

* + - 1. Control maven project using command line option
         1. open cmd and check for

mvn - - version

* + - * 1. download maven and add it to environment path

MAVEN\_HOME=/Users/balaji/Documents/Component/apache-maven-3.8.4

Under Path variable = /Users/balaji/Documents/Component/apache-maven-3.8.4/bin

JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0\_221.jdk/Contents/Home

* + - * 1. open new command prompt anc check

mvn - - version

* + - * 1. Open cmd and goto the directory of pom.xml

mvn clean compile test

* + - 1. Jenkins -
         1. get the jenkins url and login to jenkins

or

* + - * 1. start the jenkins locally to try and practice
* Download the [latest stable Jenkins WAR file](https://www.jenkins.io/download) to an appropriate directory on your machine.
* Open up a terminal/command prompt window to the download directory.
* Run the command java -jar jenkins.war.
* Browse to http://localhost:8080 and wait until the Unlock Jenkins page appears.
* Continue on with the [Post-installation setup wizard](https://www.jenkins.io/doc/book/installing/war-file/#setup-wizard) below.
* goto http://localhost:8080/

to trigger in different

* java -jar jenkins.war --httpPort 9090

20. Selenium Grid

Hub - register one machine (which is having framework)

* goto the directory of selenium jar and open cmd

java -jar selenium-server-4.1.2.jar hub

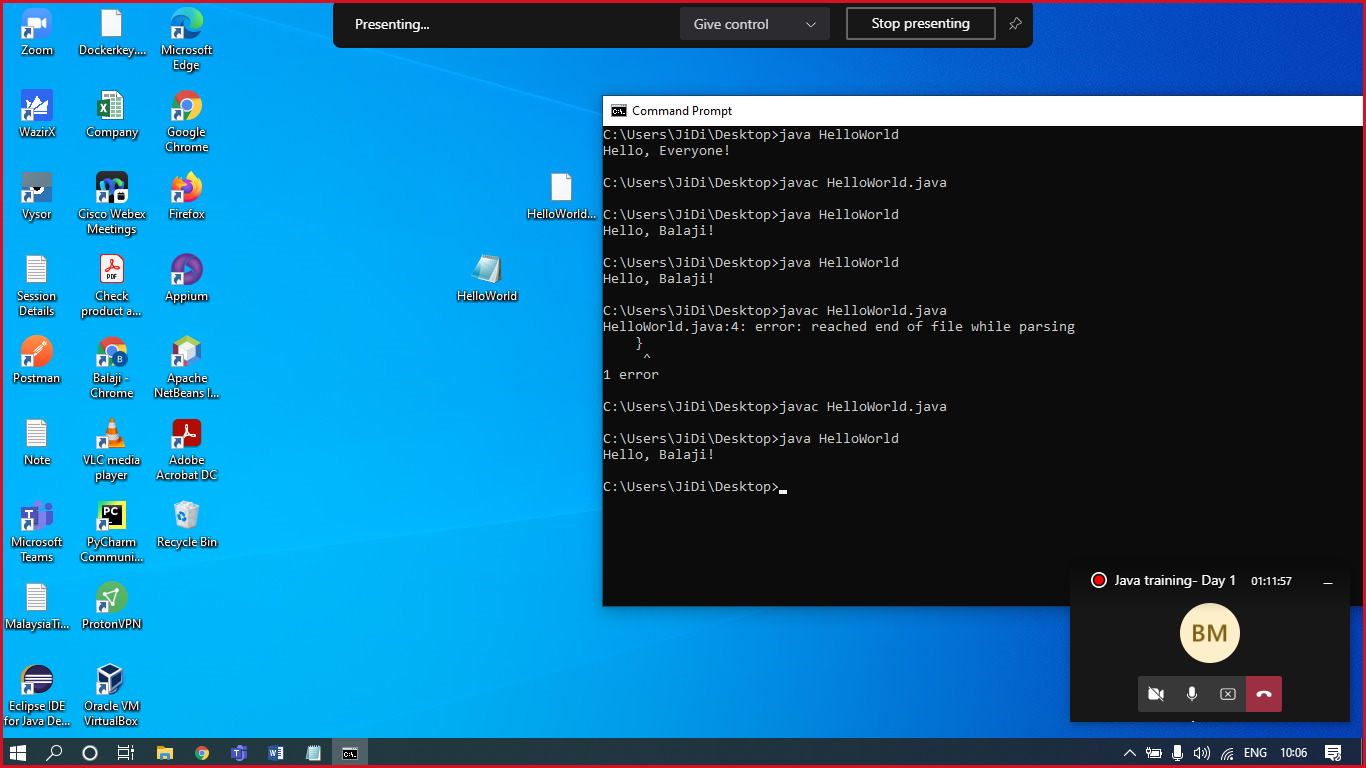
node 1 - first machine register

java -jar selenium-server-4.1.2.jar node

node 2 - second machine register

Reference:

Ref1: Img 1:



Ref2:

1GB - 1024 MB

1 MB - 1024 KB

1KB - 1024 B

1 B - 8 bits

1 bit (either 0 and 1)

Ref3:

<https://edabit.com/>

Queries:-

1. Run failed scenario

Task 1:

| **PERCENTAGE** | **GRADE** |
| --- | --- |
| 90 and above | A |
| 80 to 89 | B |
| 60 to 79 | C |
| below 60 | F |

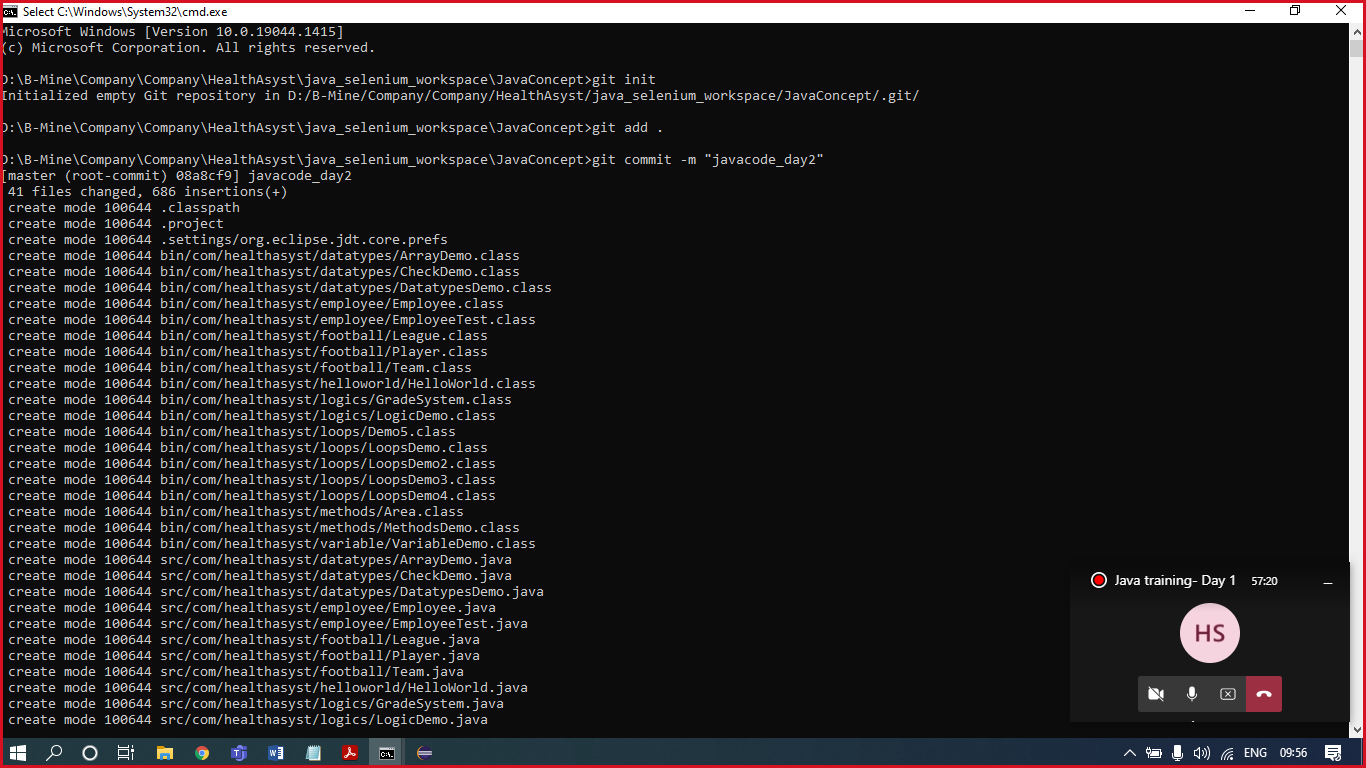
**int** i=1;

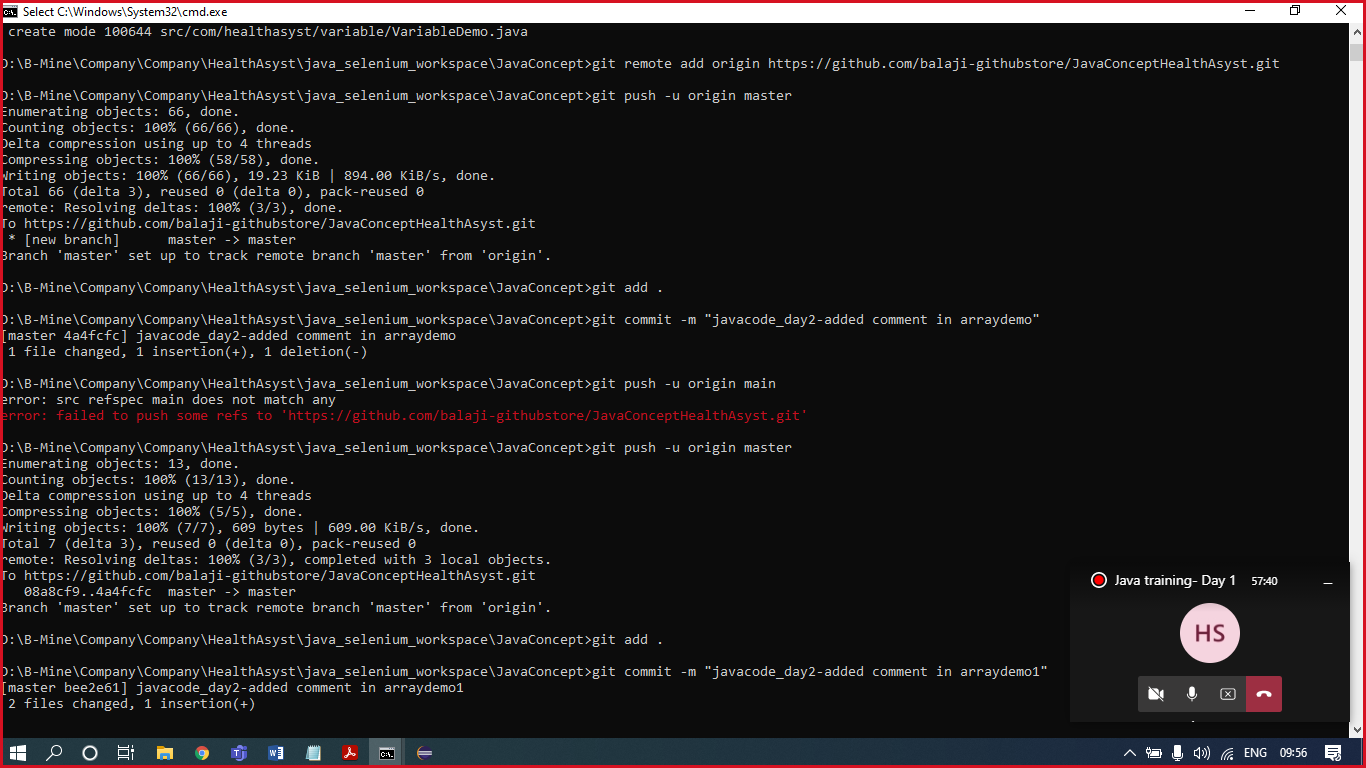
i++; // or i++ // i=i+1

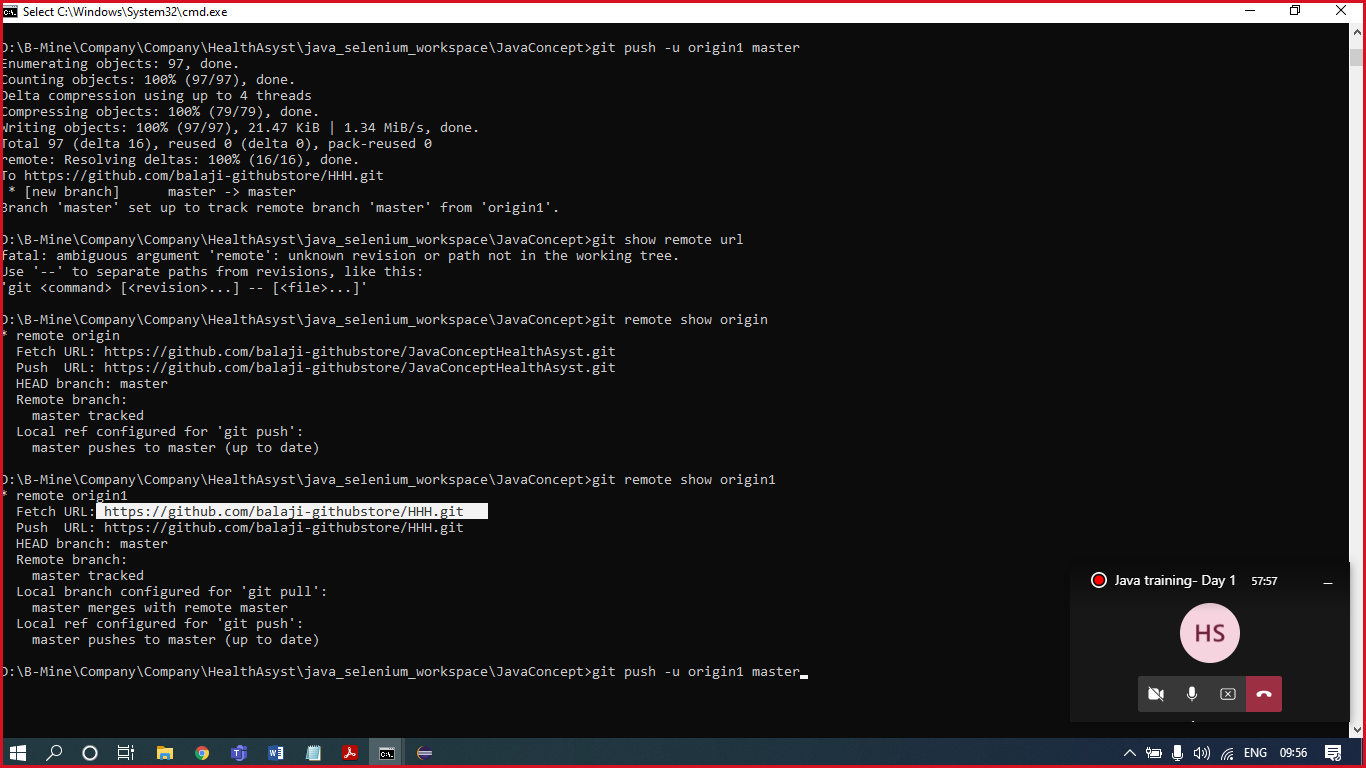
System.***out***.println(i);

Template for Student

Git:







<https://github.com/SeleniumHQ/selenium.git>

@high

Scenario: Add Patient

Given I have browser with OpenEMR page

When I enter username as 'admin'

And I enter password as 'pass'

And I select language as 'English (Indian)'

And I click on login

And I click on patient-client

And I click on patient

And I click on Add New Patient

And I fill the form

| firstname | lastname | dob | gender |

| John | Wick | 2022-02-05 | Male |

And I click on create new patient

And I click on confirm create new patient

And I store the text and handle the alert box

And I click on happy birthday if available

Then the alert text should contains 'Tobacco'

And I should get the added patient name 'Medical Record Dashboard - John Wick'

@high

Scenario: Add Patient

Given I have browser with OpenEMR page

When I enter username as 'admin'

And I enter password as 'pass'

And I select language as 'English (Indian)'

And I click on login

And I click on patient-client

And I click on patient

And I click on Add New Patient

And I fill the form

| firstname | lastname | dob | gender |

| Peter | Ken | 2022-02-05 | Male |

And I click on create new patient

And I click on confirm create new patient

And I store the text and handle the alert box

And I click on happy birthday if available

Then the alert text should contains 'Tobacco'

And I should get the added patient name 'Medical Record Dashboard - John Wick'