
Foundation Level for Test Automation

Level 1



1159 Sonora Court Suite 205,
Sunnyvale CA, 94086 USA Tel. +1
(408) 220 9912 Ext 1
www.qauber.com

Who am I?



- Yagnanarayana Dande - BE from BITS Pilani <https://www.linkedin.com/in/yagnanarayana-dande-1876777>
- 10 Years in Software Industry, Filed a Patent Application with Symantec and got Trade Secret for the invention
- 7 Years in Big Data & Analytics, 6 Years in Test Automation
- Technical Review Board Member for Packt Publishing - Vertica, TestNG
- Certified Java Programmer, ISTQB Certified Tester, Part of Hadoop Core Team
- Vivid blogger at <http://qabypassion.blogspot.com>
- CoFounder of Elephant Tamers

Introduction



1159 Sonora Court Suite 205,
Sunnyvale CA, 94086 USA Tel. +1
(408) 220 9912 Ext 1
www.qauber.com

Introduction

Course Format

4 weeks, 2 times a week, 2 hr classes + lab, homework, exams

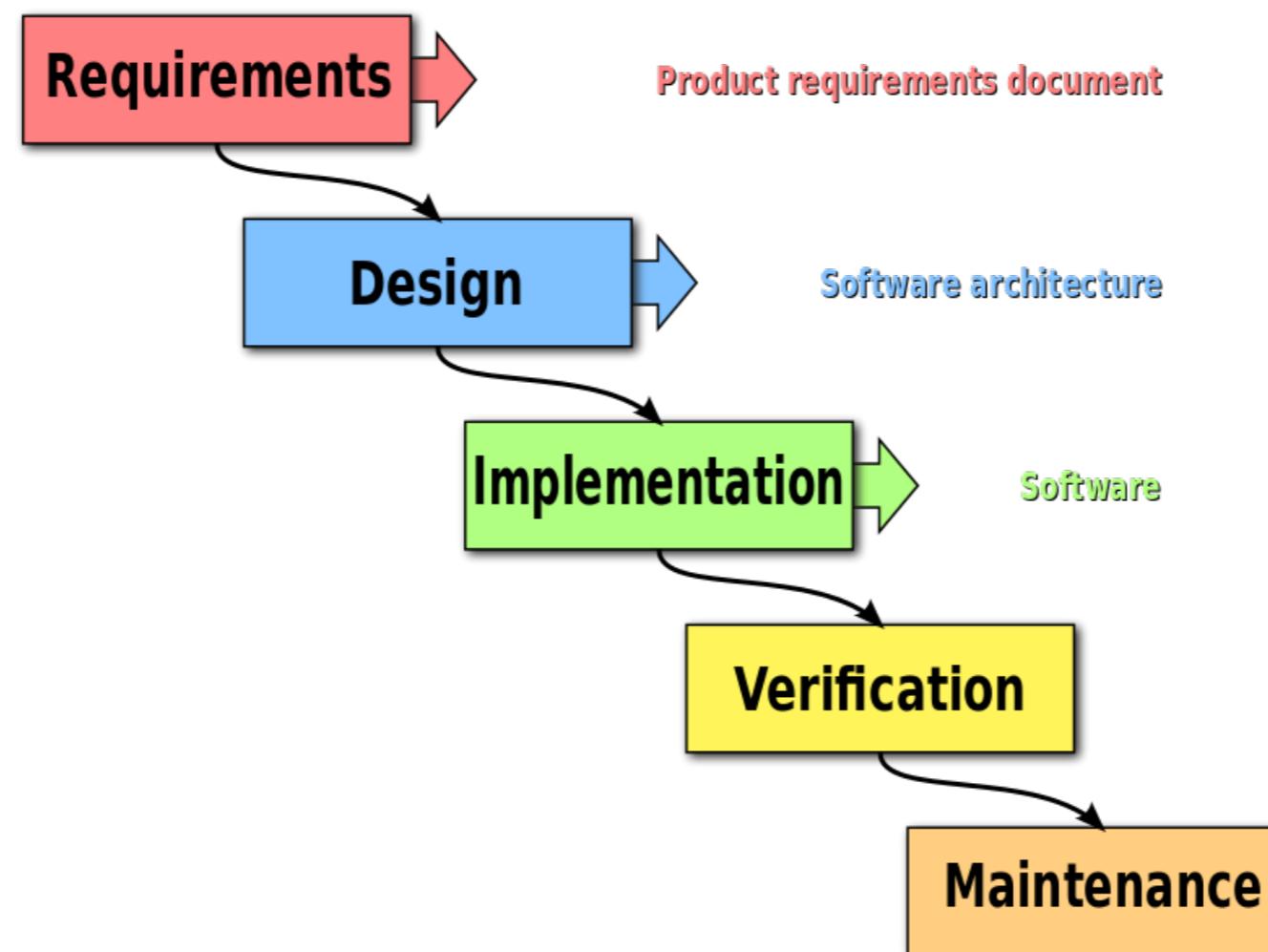
Course Content

[https://docs.google.com/document/d/
1N32tZW20NSg6fLPpHy2nCb9qX7MtCyp3LOLJ73gA0Yk/edit](https://docs.google.com/document/d/1N32tZW20NSg6fLPpHy2nCb9qX7MtCyp3LOLJ73gA0Yk/edit)

Software Development Life Cycle (SDLC)

- Help produce a product that is cost-efficient, effective, and of high quality.
- SDLC creates a structure for the development teams by defining various tasks that need to happen.
- The methodology within the SDLC process can vary across organizations.

Software Development Life Cycle (SDLC)



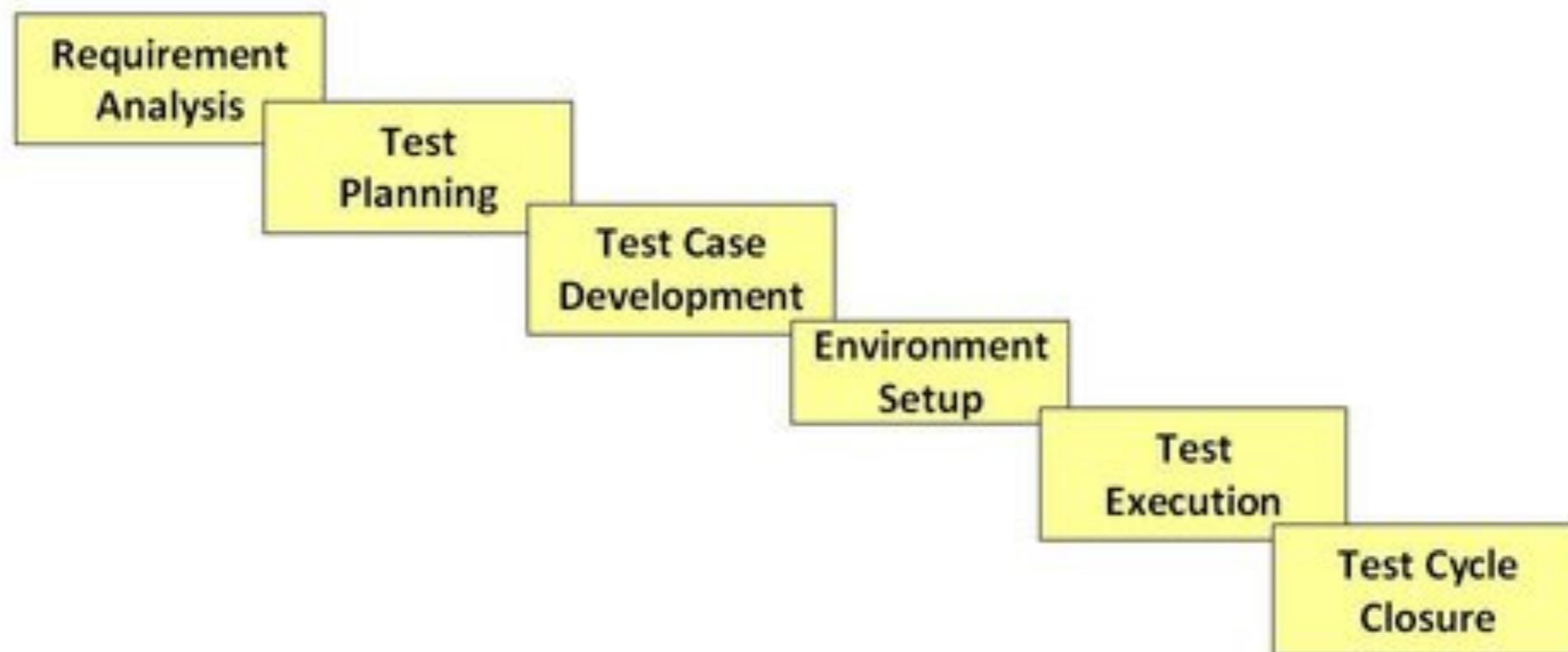
Software Test Life Cycle (STLC)

Software Testing Life Cycle (STLC) defines sequence of activities performed to certify software product through software Testing.

Each of these stages have a definite Entry and Exit criteria; , Activities & Deliverables associated with it.

In an Ideal world you will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible

Software Test Life Cycle (STLC)



Requirements Analysis

During this phase, test team studies the requirements from a testing point of view to identify the testable requirements.

The QA team may interact with various stakeholders (Client, Business Analyst, Technical Leads, System Architects etc) to understand the requirements in detail.

Requirements could be either Functional (defining what the software must do) or Non-Functional (defining system performance /security)

Automation feasibility for the given testing project is also done in this stage.

Test Planning

This phase is also called Test Strategy phase.

Typically , in this stage, a Senior QA member will determine effort and cost estimates for the project and would prepare and finalize the Test Plan.

- Preparation of test plan/ test strategy document
- Test tool selection
- Test effort estimation
- Resource planning and determining roles and responsibilities.
- Training requirements

Test Case Development

This phase involves creation, verification and rework of test cases & test scripts. Test data , is identified/created and is reviewed and then reworked as well.

- * Create test cases, automation scripts (if applicable)
- * Review test cases and scripts
- * Create test data

Test Environment Setup

Test environment decides the software and hardware conditions under which a work product is tested.

Test environment setup is one of the critical aspects of testing process and can be done in parallel with Test Case Development Stage.

Test Execution

During this phase test team will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.

- * Execute tests as per plan
- * Document test results, and log defects for failed cases
- * Map defects to test cases
- * Retest the fixed defects
- * Track the defects to closure

Test Cycle Closure

Testing team will meet , discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle.

The idea is to remove the process bottlenecks for future test cycles and share best practices for any similar projects in future.

Software Development Models

- **Waterfall model** is a sequential (non-iterative) design process in which progress is seen as flowing like a waterfall through the phases.
- **Agile model** is a models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

Software Applications

- Desktop Applications
 - Office Products like Word, Excel, Powerpoint
- Web Applications
 - Gmail, Flipkart
- Mobile Applications
 - Facebook lite, Bank Apps

Web application and Examples

A **web based application** is a software package that can be accessed through the **web** browser. The software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.

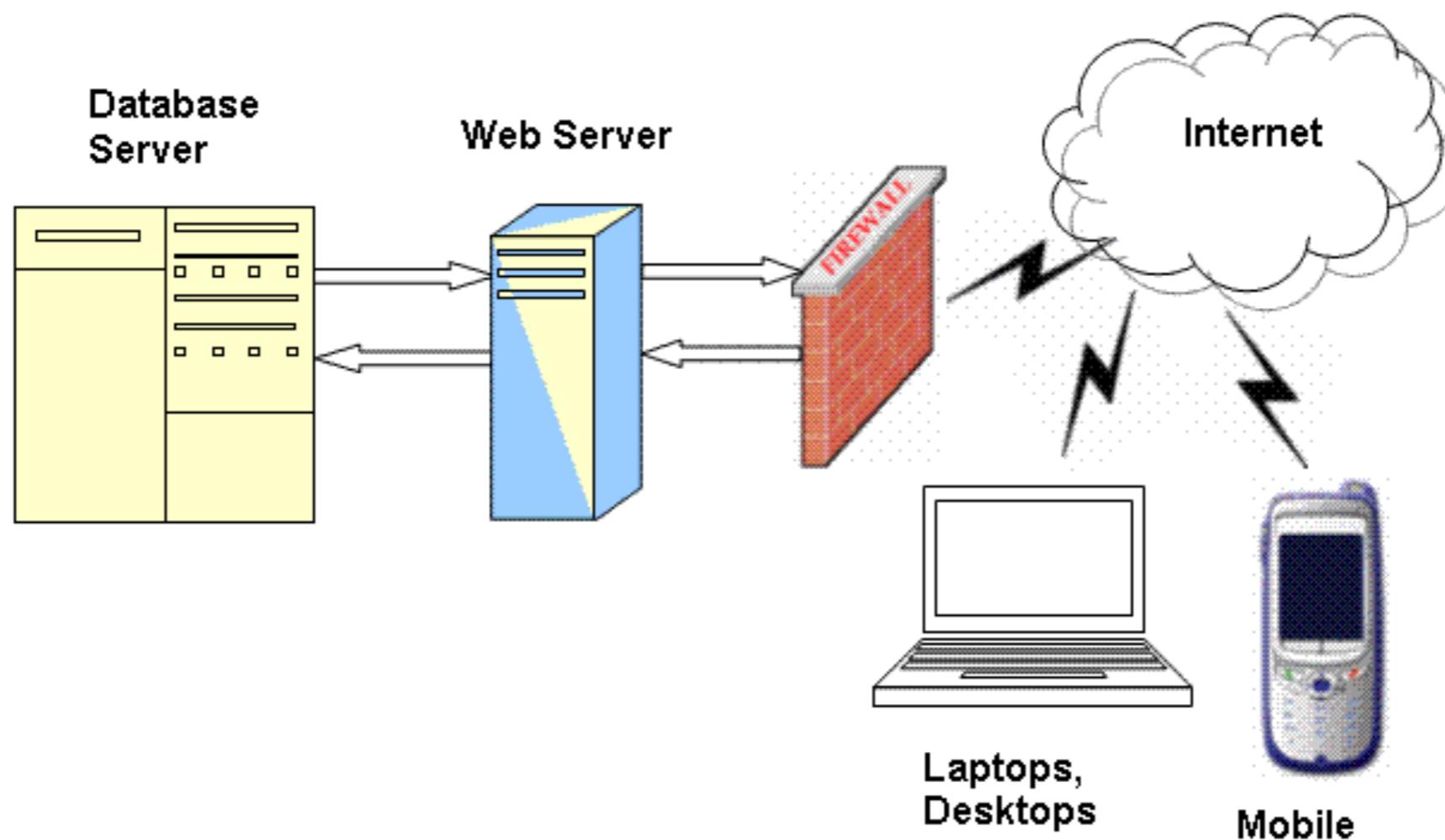
(Source: www.netsity.com)

Examples:

www.gmail.com

www.amazon.com

Web Application Architecture



Testing Web Applications

- Test all links
- Forms
- Cookies
- HTML, CSS
- Business Workflow
- Site Navigation
- Content,

Regression Testing

Whenever developers change or modify their software, even a small tweak can have unexpected consequences.

Regression testing is testing existing software applications to make sure that a change or addition hasn't broken any existing functionality.

Its purpose is to catch bugs that may have been accidentally introduced into a new build or release candidate, and to ensure that previously eradicated bugs continue to stay dead.

Automating Regression Tests

By re-running testing scenarios that were originally scripted when known problems were first fixed, you can make sure that any new changes to an application haven't resulted in a regression, or caused components that formerly worked to fail.

Such tests can be performed manually on small projects, but in most cases repeating a suite of tests each time an update is made is too time-consuming and complicated to consider.

Hence we automate them to reduce time and give faster feedback.

Test Automation for Web Application

Automate Operations that can be performed by Users on Web Application through browser with different Data Sets.

Examples:

Login in www.gmail.com -> Compose Mail -> Logout

Data Set: Different Credentials, Different Content Types

Login in www.amazon.com -> Checkout Items -> Buy them -> Logout

Data Set: Different Credentials, Different Products

Tools to Automate Web Application Testing

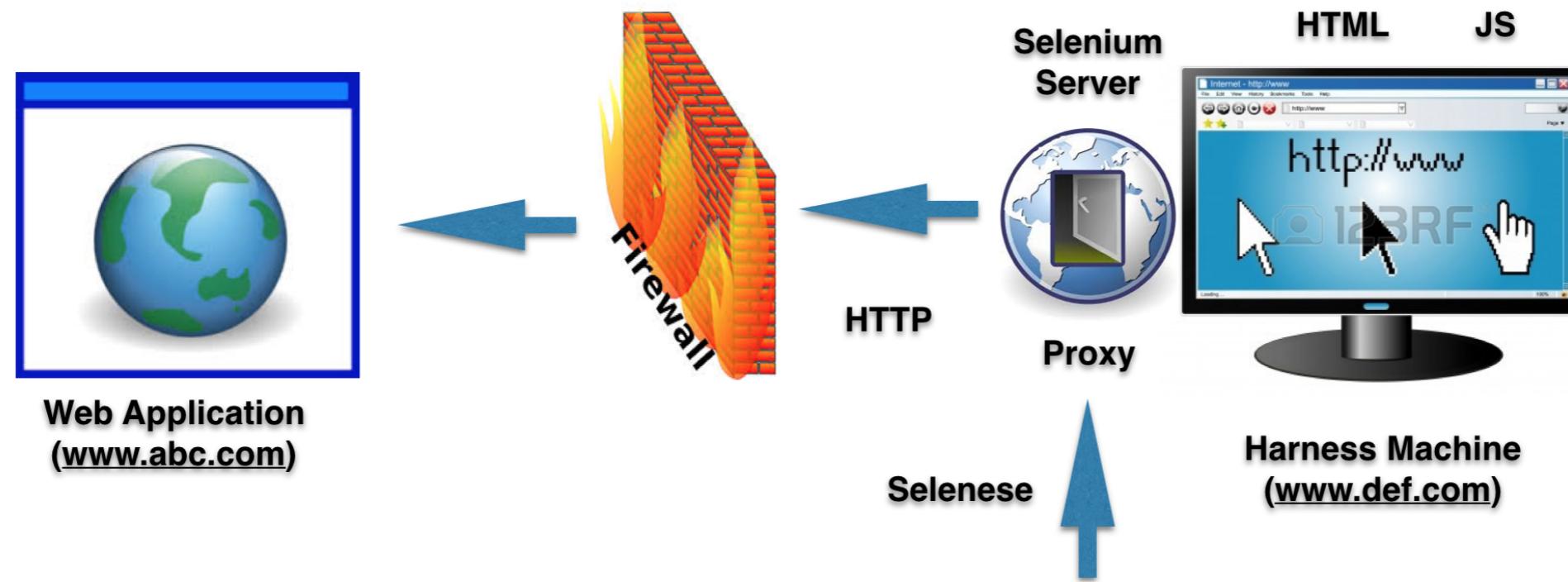
- Sahi , WatiR, QTP, Sikuli, Selenium, etc.. are available
- Selenium is more popular as its FREE, supported in various browsers like IE, Firefox, Chrome, Safari, Opera, Edge and allows to script in various languages like Ruby, Java, PHP, Perl, Python, C#, Groovy.

Selenium IDE

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. It's an easy-to-use Firefox plug-in.

This allows us to Record and Playback the Tests

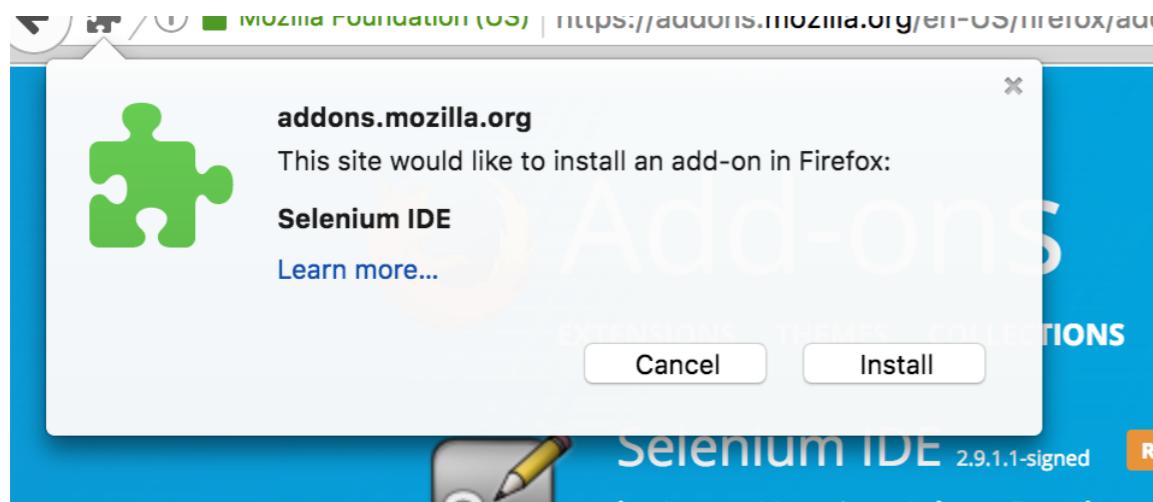
How Selenium IDE Works



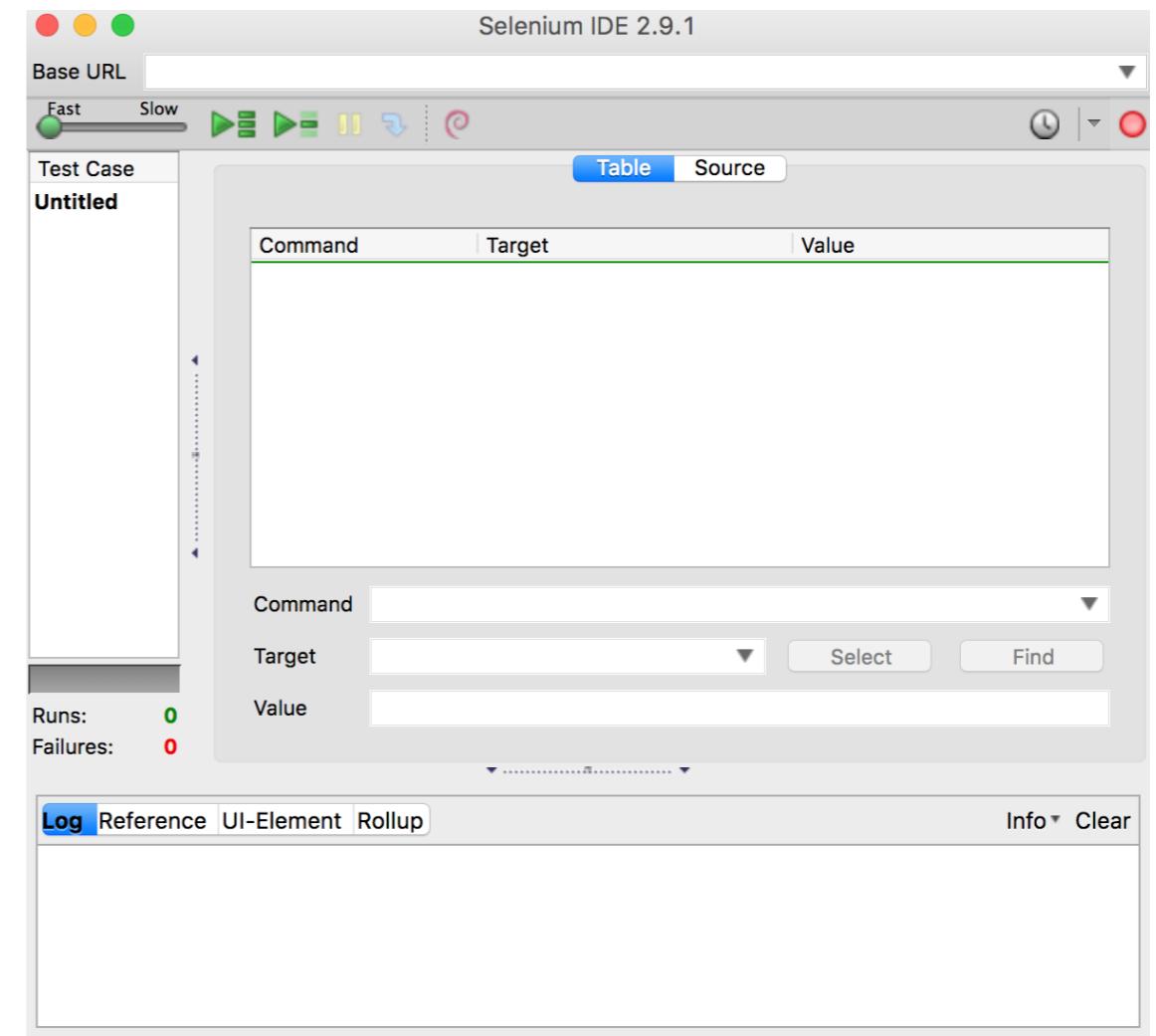
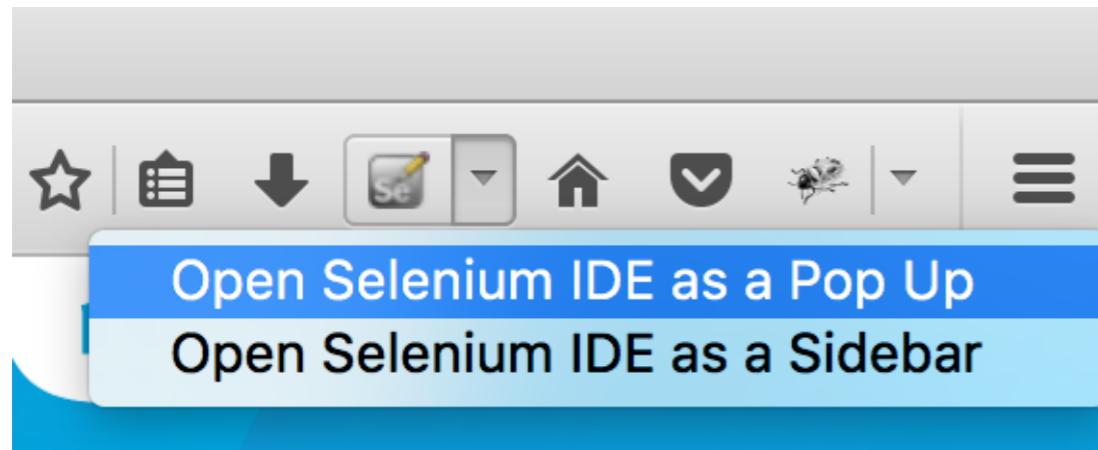
Location	Action	Value1	Value2
Browser	Open	www.gmail.com	
Email Box	Type Text	test@gmail.com	
Password Box	Type Text	#2	
Submit Button	Click		

Installing Selenium IDE

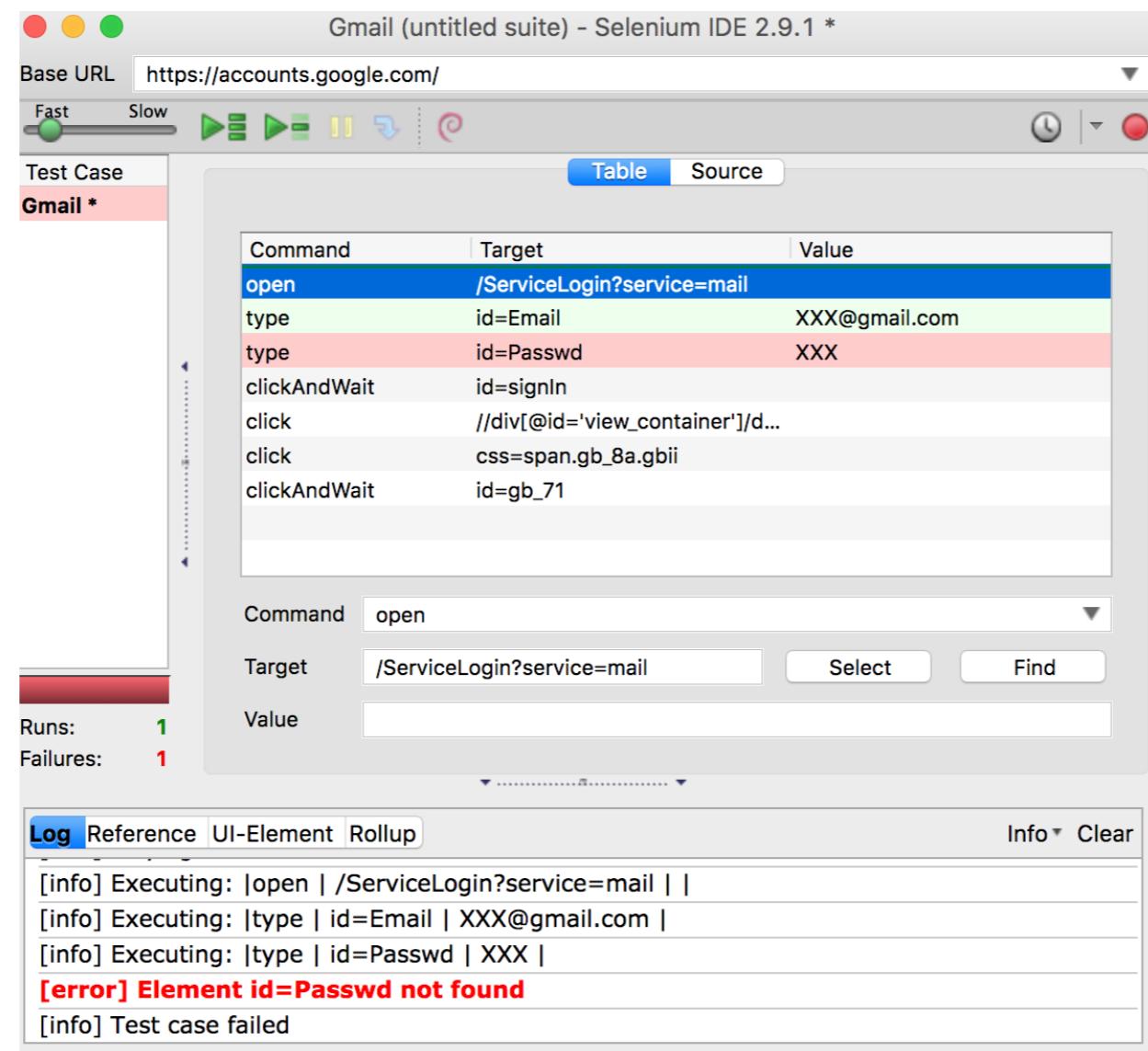
<https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>



Open IDE



Selenium IDE for test automation

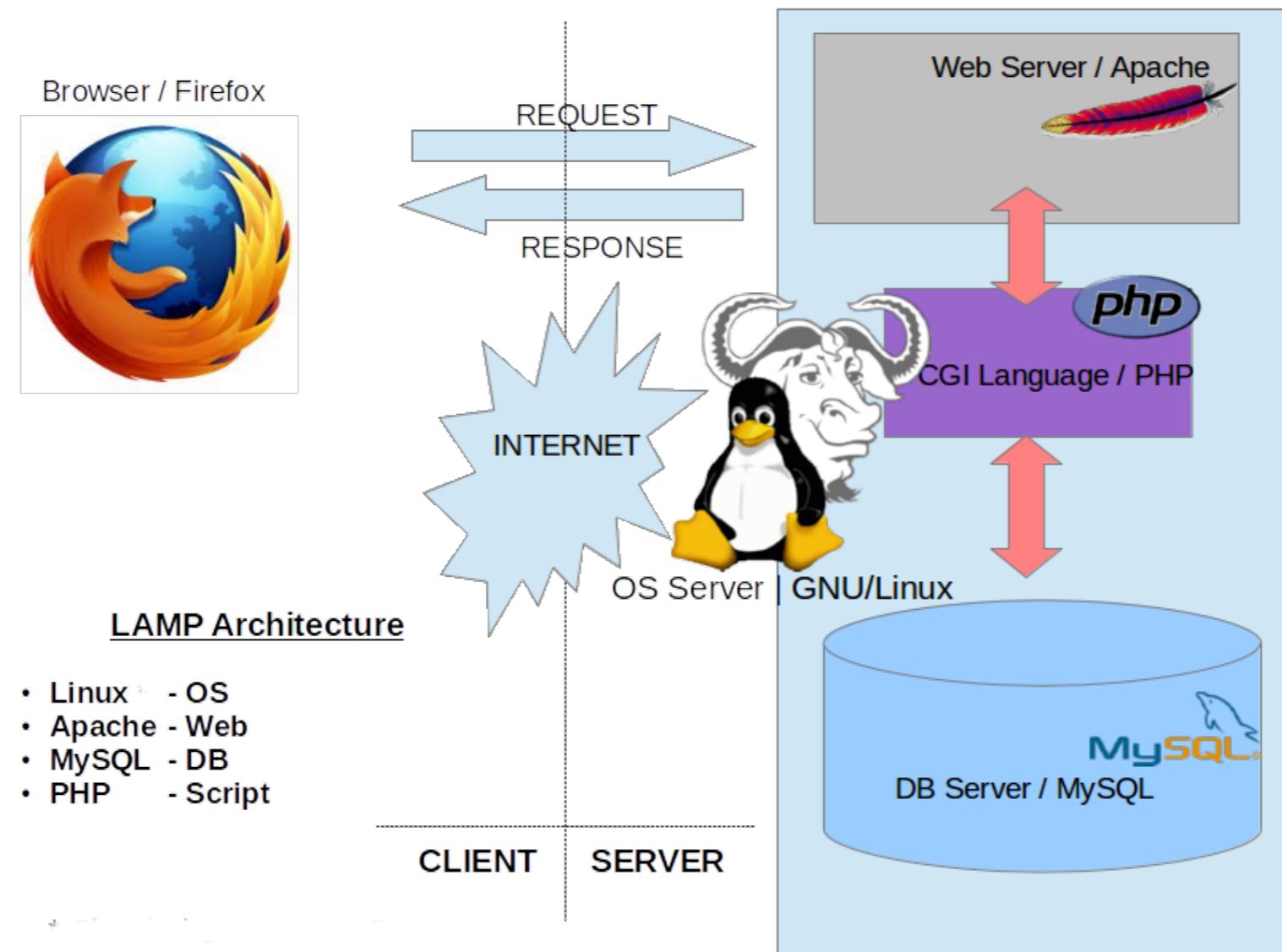


Web Application Basic Architecture and Automation



**1159 Sonora Court Suite 205,
Sunnyvale CA, 94086 USA** Tel. +1
(408) 220 9912 Ext 1
www.qauber.com

LAMP



knowing about LAMP server

LAMP stands for Linux, Apache, MySQL, PHP

Linux is an Operating System

Apache is WebServer

MySQL is Database Server

PHP is scripting language that is especially suited for web development

Operating System

Some examples include

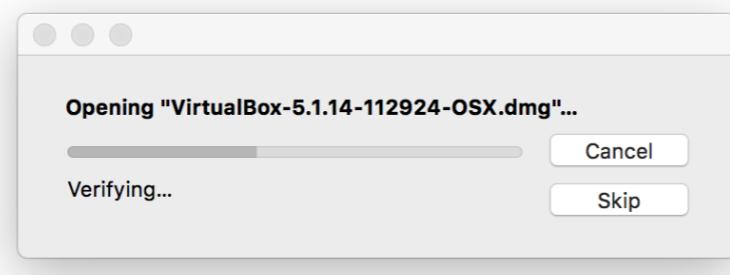
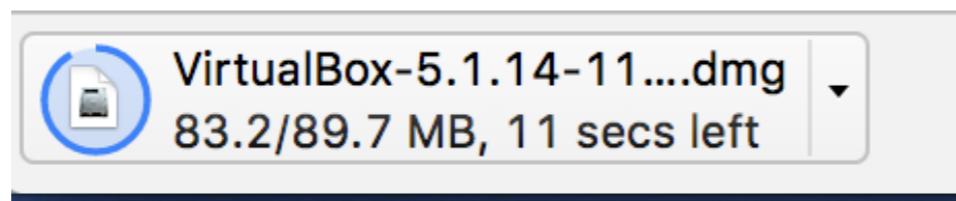
- Microsoft Windows like Windows 10, Windows 8, Windows 7, Windows Vista, and Windows XP)
- Apple's macOS (formerly OS X)
- Chrome OS
- Flavors of the open source operating system Linux like Ubuntu, CentOS

Physical and Virtual Machines

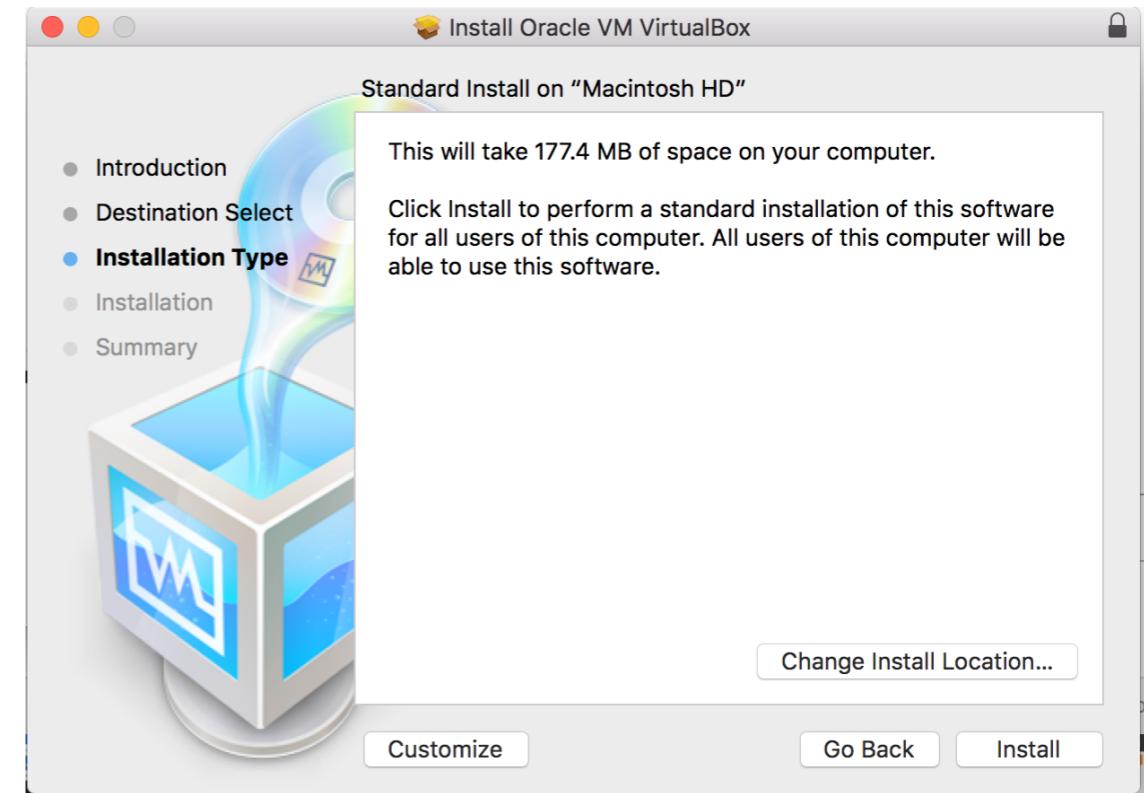
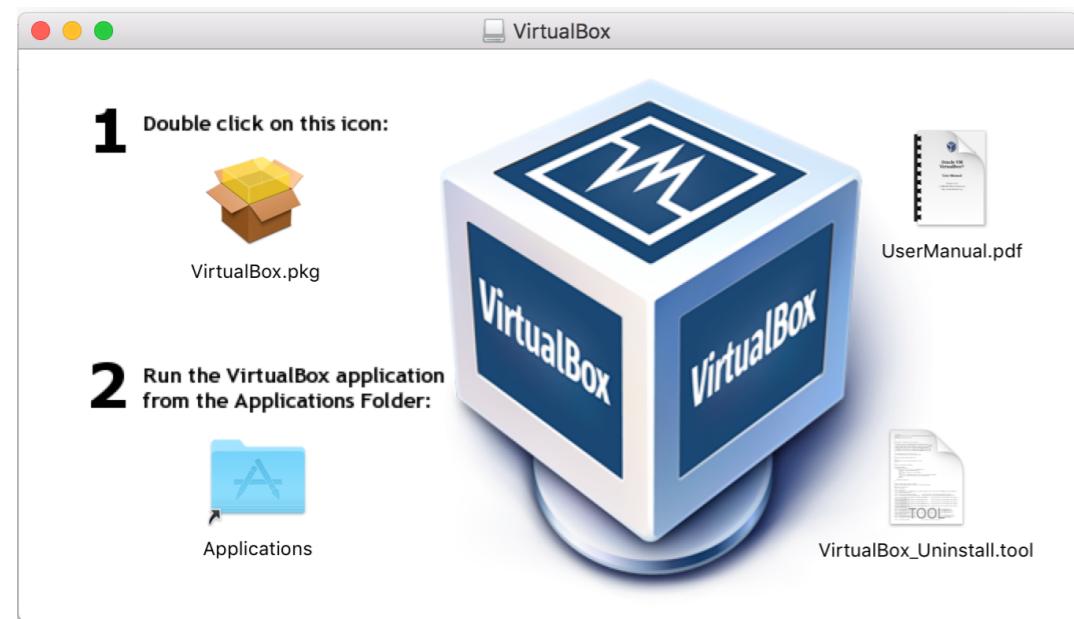
- Physical Machines is your laptop, desktop or lab machine. It has all physical components like CPU, RAM, HardDisk
- Virtual Machine is a machine sitting virtually on top of your Physical Machine.
Example: Machine created on VMWare VirtualBox.
- Virtual Machine shares physical hardware meaning you can create several virtual machines on a single Physical Box
- Physical Machine which hosts Virtual Machine is called as Host and Virtual Machine is a Guest Machine

Installation of Oracle Virtual Box

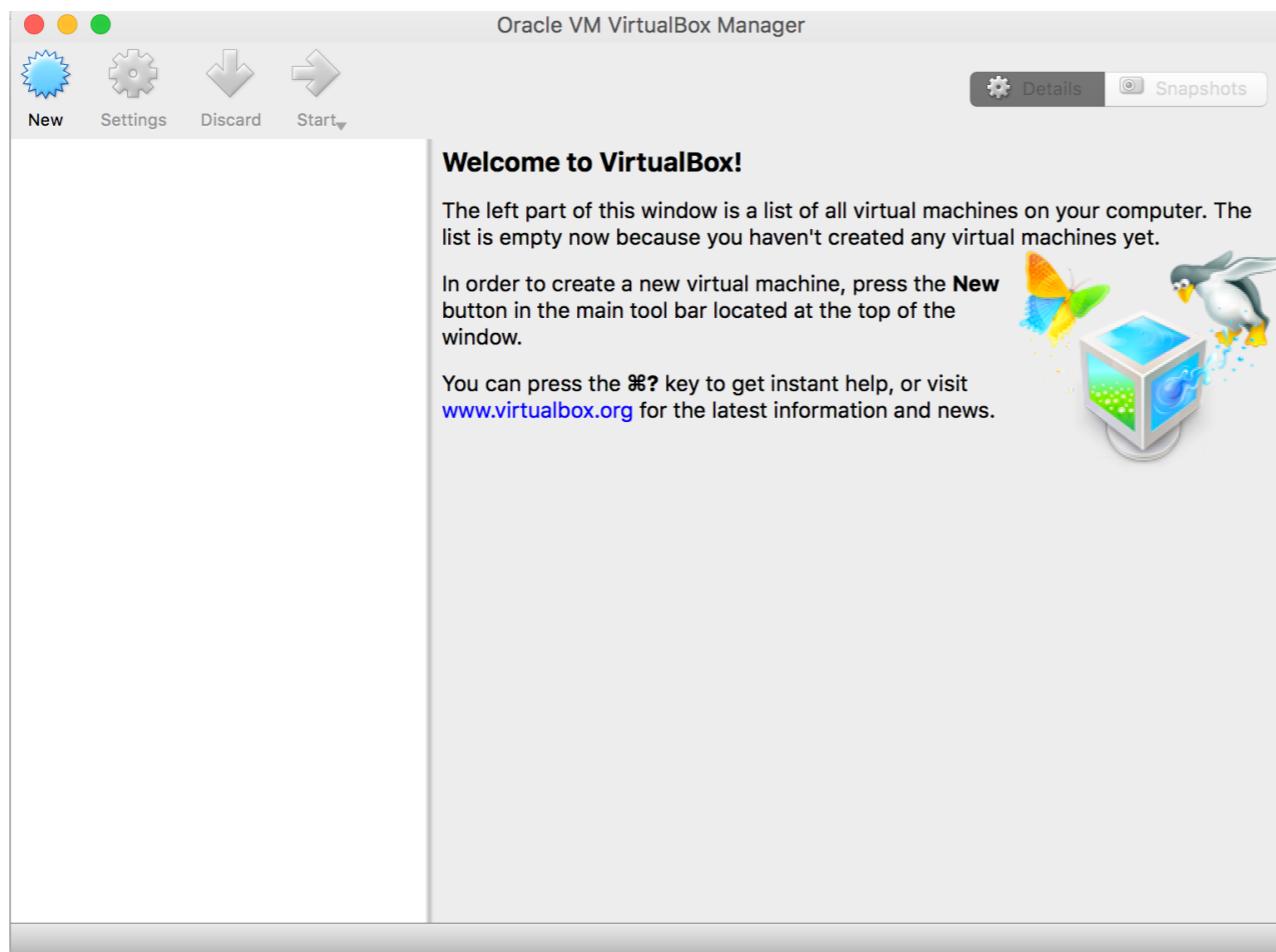
- Download from <http://download.virtualbox.org/virtualbox/5.1.14/VirtualBox-5.1.14-112924-OSX.dmg>



Installation of Oracle Virtual Box



Installation of Oracle Virtual Box



Vagrant Installation

Download Vagrant

<http://www.vagrantup.com/downloads>



The screenshot shows the Vagrant download page. On the left, there's a sidebar with a blue header "DOWNLOAD" containing links for "Latest" and "Old Versions". The main content area has a light gray background. At the top, there are navigation links: "VMWARE INTEGRATION" (highlighted in blue), "DOWNLOADS", "DOCUMENTATION", and "BL". The main section is titled "DOWNLOAD VAGRANT" and contains text about available downloads for Vagrant 1.6.3. It lists three download options: "MAC OS X Universal (32 and 64-bit)", "WINDOWS Universal (32 and 64-bit)", and "LINUX (DEB) 32-bit | 64-bit". Each option includes its respective operating system logo.

VMWARE INTEGRATION DOWNLOADS DOCUMENTATION BL

DOWNLOAD VAGRANT

Below are all available downloads for the latest version of Vagrant (1.6.3). Please download the proper package for your operating system and architecture. You can find SHA256 checksums for packages [here](#).

 MAC OS X
[Universal \(32 and 64-bit\)](#)

 WINDOWS
[Universal \(32 and 64-bit\)](#)

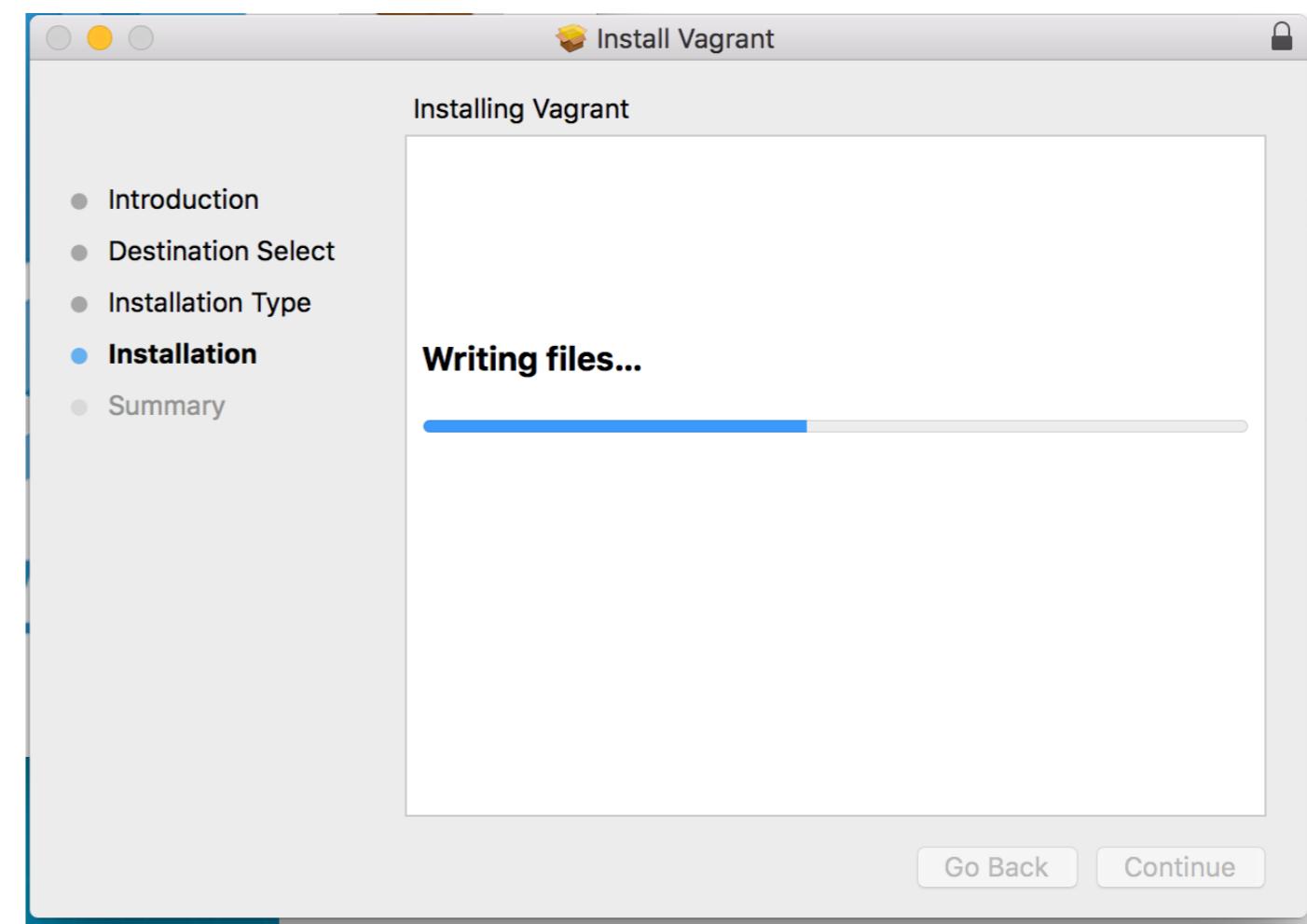
 LINUX (DEB)
[32-bit](#) | [64-bit](#)

Installing Vagrant

Installing Vagrant is extremely easy. Download and install it using standard procedures for your operating system.

The installer will automatically add vagrant to your system path so that it is available in terminals.

Installing Vagrant



Deploy CentOS Machine using Vagrant

```
mkdir -p ~/vagrant/centos7 (mkdir vagrant\centos for Windows)
```

```
cd ~/vagrant/centos7 (cd vagrant\centos7 for Windows)
```

cygwin is required for Windows for rsync and putty.exe for ssh

```
vagrant init centos/7; vagrant up --provider virtualbox
```

Above Step will take a lot of time as it will download files from internet

```
vagrant status
```

```
vagrant ssh
```

Unix Command - ls

The ls command - the list command - functions in the Linux terminal to show all of the major directories that are under a given file system. For example, the command:

```
ls /applications
```

...will show the user all of the folders stored in the overall applications folder.

The ls command is used for viewing files, folders/directories.

Linux Command - cd

The cd command - change directory - will allow the user to change between file directories. As the name command name suggest, you would use the cd command to circulate between two different directories.

```
cd /hello
```

Linux Command - cp

The cp command - copy - allows a user to copy a file to another folder or directory.

Linux Command - mv

The mv command - move - allows a user to move a file to another folder or directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents" folder, the mv command functions in the same manner.

Linux Command - man

The man command - the manual command - is used to show the manual of the inputted command. Just like a film on the nature of film, the man command is the meta command of the Linux CLI. Inputting the man command will show you all information about the command you are using.

Linux Command - mkdir

The mkdir - make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the mkdir command makes new directories in a Linux environment.

Linux Command - rmdir

The rmdir - remove directory - command allows the user to remove an existing command using the Linux CLI.

Linux Command - touch

The touch command - a.k.a. the make file command - allows users to make files using the Linux CLI. Just as the mkdir command makes directories, the touch command makes files. Just as you would make a .doc or a .txt using a PC desktop, the touch command makes empty files.

Linux Command - rm

The rm command - remove - like the rmdir command is meant to remove files from your Linux OS. Whereas the rmdir command will remove directories and files held within, the rm command will delete created files

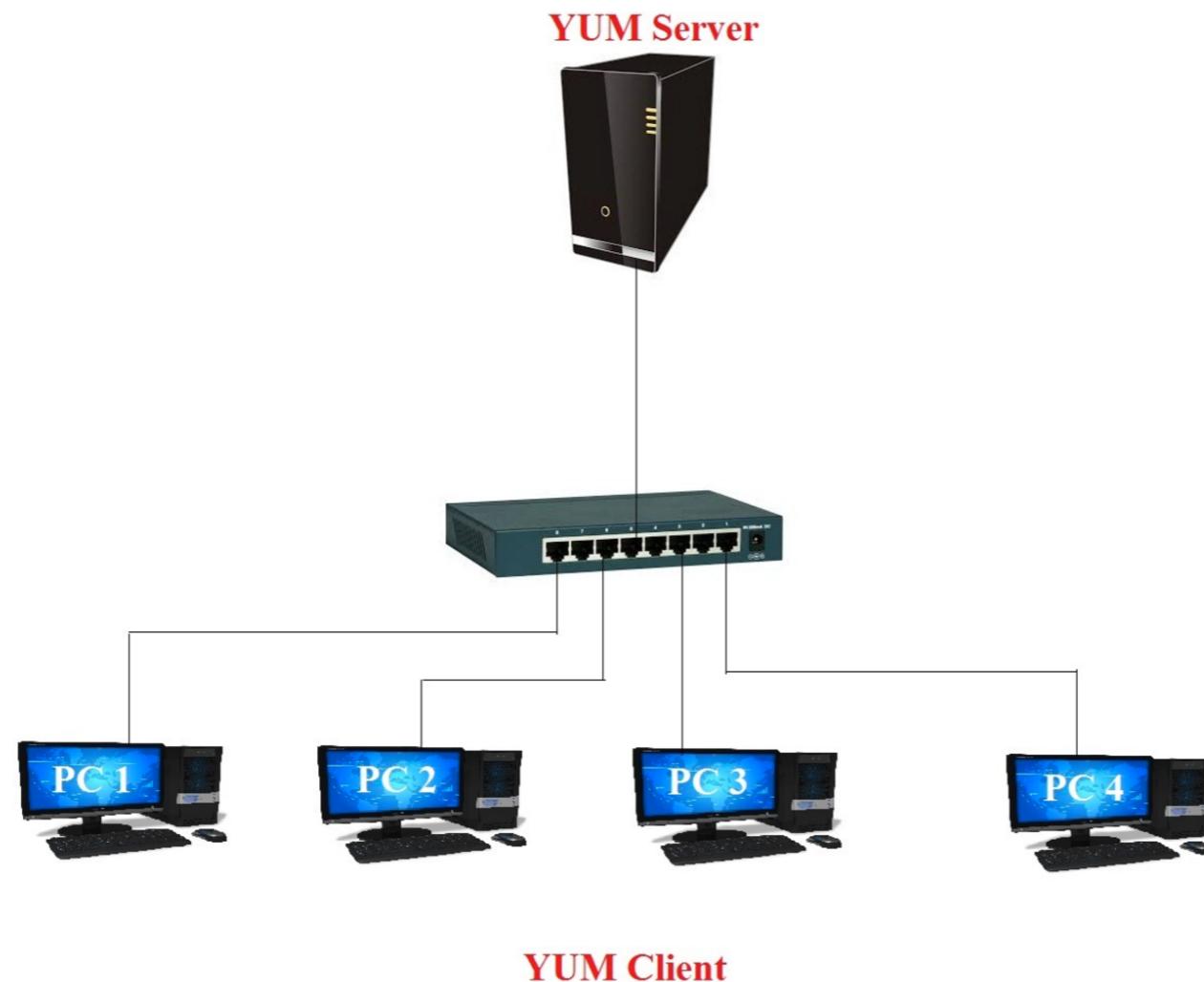
Linux Command - clear

The clear command does exactly what it says. When your Linux CLI gets all mucked up with various readouts and information, the clear command clears the screen and wipes the board clean. Using the clear command will take the user back to the start prompt of whatever directory you are currently operating in.

Unix Commands

- In CentOS, applications are installed through rpms
- In CentOS rpms can be installed using installer called yum
- yum will install these rpms from a centralized location called yum repository

Installing Applications in CentOS



Web Server

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients.

Install Apache

sudo su

```
yum install httpd httpd-devel  
service httpd start
```

in VagrantFile Uncomment config.vm.network "forwarded_port", guest: 80, host: 8080

vagrant reload

To make sure everything installed correctly we will now test Apache to ensure it is working properly.

Open up any web browser on your laptop and then enter the following into the web address:
<http://127.0.0.1:8080/>

You should see a folder entitled apache2-default/. Open it and you will see a message saying "It works!" , congrats to you!

Database Server

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated.

Examples: Oracle, MS SQL, mysql

Install MySQL

```
sudo su
```

```
yum install mysql mysql-devel wget
```

```
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
```

```
rpm -ivh mysql-community-release-el7-5.noarch.rpm
```

```
yum update
```

```
yum install mysql-server
```

```
systemctl start mysqld
```

Setup MySQL

mysql

```
mysql> USE mysql;  
mysql> UPDATE user SET Password=PASSWORD('newpassword') WHERE  
user='root';  
mysql> FLUSH PRIVILEGES;
```

```
mysql>quit;
```

```
mysql -u root -p  
Enter Password: newpassword
```

DB Commands

SHOW DATABASES

CREATE DATABASE

USE

SHOW TABLES

SELECT

Create Tables for Sample Web Application

```
mysql> create database test;
```

```
mysql> use test;
```

```
mysql> CREATE TABLE IF NOT EXISTS `user` (
    ->   `id` int(11) NOT NULL AUTO_INCREMENT,
    ->   `username` varchar(255) NOT NULL,
    ->   `email` varchar(255) NOT NULL,
    ->   `password` varchar(255) NOT NULL,
    ->   PRIMARY KEY (`id`),
    ->   UNIQUE KEY `username` (`username`)
    -> );
```

```
mysql> select * from user;
Empty set (0.00 sec)
```

```
mysql>quit;
```

Install PHP

```
yum install php php-mysqlnd php-common php-gd php-  
mbstring php-mcrypt php-devel php-xml
```

```
service httpd restart
```

Test PHP

Create a file named /var/www/html/test.php with the following phpinfo() function inside php quotes.

```
// test.php
```

```
<?php  
phpinfo();  
?>
```

Then point your browser to <http://127.0.0.1:8080/test.php>

That's it! You should see a php configuration file displaying all kind of paths and installed modules.

Application Running on LAMP

<http://localhost:8080/Register.php>

Please Register

logs

* HTTPD Logs at tail -f /var/log/httpd/error_log and tail -f /var/log/httpd/access_log

```
[root@localhost vagrant]# tail -f /var/log/httpd/error_log
[Thu Jan 26 09:41:12.306434 2017] [core:notice] [pid 13835] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Thu Jan 26 09:41:12.307177 2017] [suexec:notice] [pid 13835] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally to suppress this message
[Thu Jan 26 09:41:12.328115 2017] [auth_digest:notice] [pid 13835] AH01757: generating secret for digest authentication ...
[Thu Jan 26 09:41:12.328845 2017] [lbmethod_heartbeat:notice] [pid 13835] AH02282: No slotmem from mod_heartbeat
[Thu Jan 26 09:41:12.346433 2017] [mpm_prefork:notice] [pid 13835] AH00163: Apache/2.4.6 (CentOS) PHP/5.4.16 configured -- resuming normal operations
[Thu Jan 26 09:41:12.346464 2017] [core:notice] [pid 13835] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Thu Jan 26 09:41:28.901496 2017] [:error] [pid 13840] [client 10.0.2.2:57388] PHP Warning: mysqli_connect(): Headers and client library minor version mismatch. Headers:50550 Library:50635 in /var/www/html/connect.php on line 2, referer: http://localhost:8080/Register.php
[Thu Jan 26 09:41:32.644145 2017] [:error] [pid 13840] [client 10.0.2.2:57388] PHP Warning: mysqli_connect(): Headers and client library minor version mismatch. Headers:50550 Library:50635 in /var/www/html/connect.php on line 2
[Thu Jan 26 09:41:43.128276 2017] [:error] [pid 13839] [client 10.0.2.2:57397] PHP Warning: mysqli_connect(): Headers and client library minor version mismatch. Headers:50550 Library:50635 in /var/www/html/connect.php on line 2, referer: http://localhost:8080/Register.php
```

Working Web Application

User Created Successfully.

Please Register

@

Remember me

```
mysql> select * from user;
+----+-----+-----+-----+
| id | username | email        | password |
+----+-----+-----+-----+
|  1 | hi       | helo@gmail.com | sdf      |
|  2 | secondtests | test@gmail.com | 1234    |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Selenium Overview, Introduction and setting up project



1159 Sonora Court Suite 205,
Sunnyvale CA, 94086 USA Tel. +1
(408) 220 9912 Ext 1
www.qauber.com

Introduction as tool for web application test automation

WebDriver is a web automation framework that allows you to execute your tests against different browsers, not just Firefox (unlike Selenium IDE).

WebDriver also enables you to use a programming language in creating your test scripts (not possible in Selenium IDE).

You can now use conditional operations like if-then-else or switch-case

You can also perform looping like do-while.

You can connect to Databases to test Results.

Following programming languages are supported by WebDriver

Java, .Net, PHP, Python, Perl, Ruby

Cross browser testing

Cross browser testing is to test your website or application in multiple browsers- and making sure that it works consistently and as intended without any dependencies, or compromise in Quality.



Cross Browser Testing using Selenium WebDriver

- Run Same Test Case using different Drivers (Browsers)
- This can be achieved using Selenium WebDriver as it supports almost all the browsers
- Also this can be achieved using Selenium Grid

Setting up Java

Download

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Double click it

Setting up Eclipse

Download <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/mars2>

Extract and Open Eclipse

Setting up Gradle (Build Tool)

Download <https://gradle.org/gradle-download/>

Extract it and Put bin/gradle in the PATH

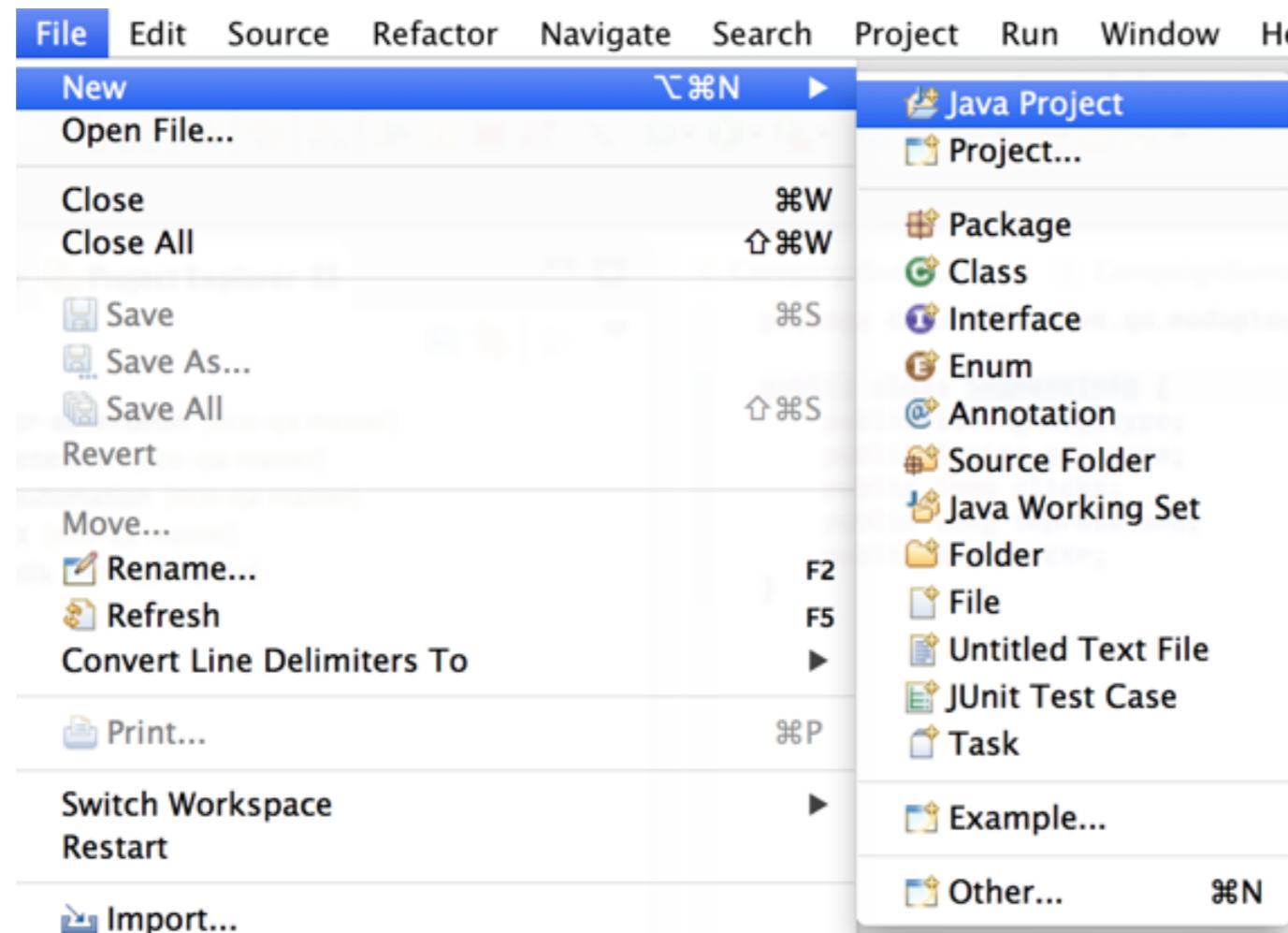
Creating and running tests on the setup

- Create Folder Structure as follows
- Project Folder -> src ->main -> java
- Create build.gradle file in Project Folder
- Run gradle cleanEclipse eclipse

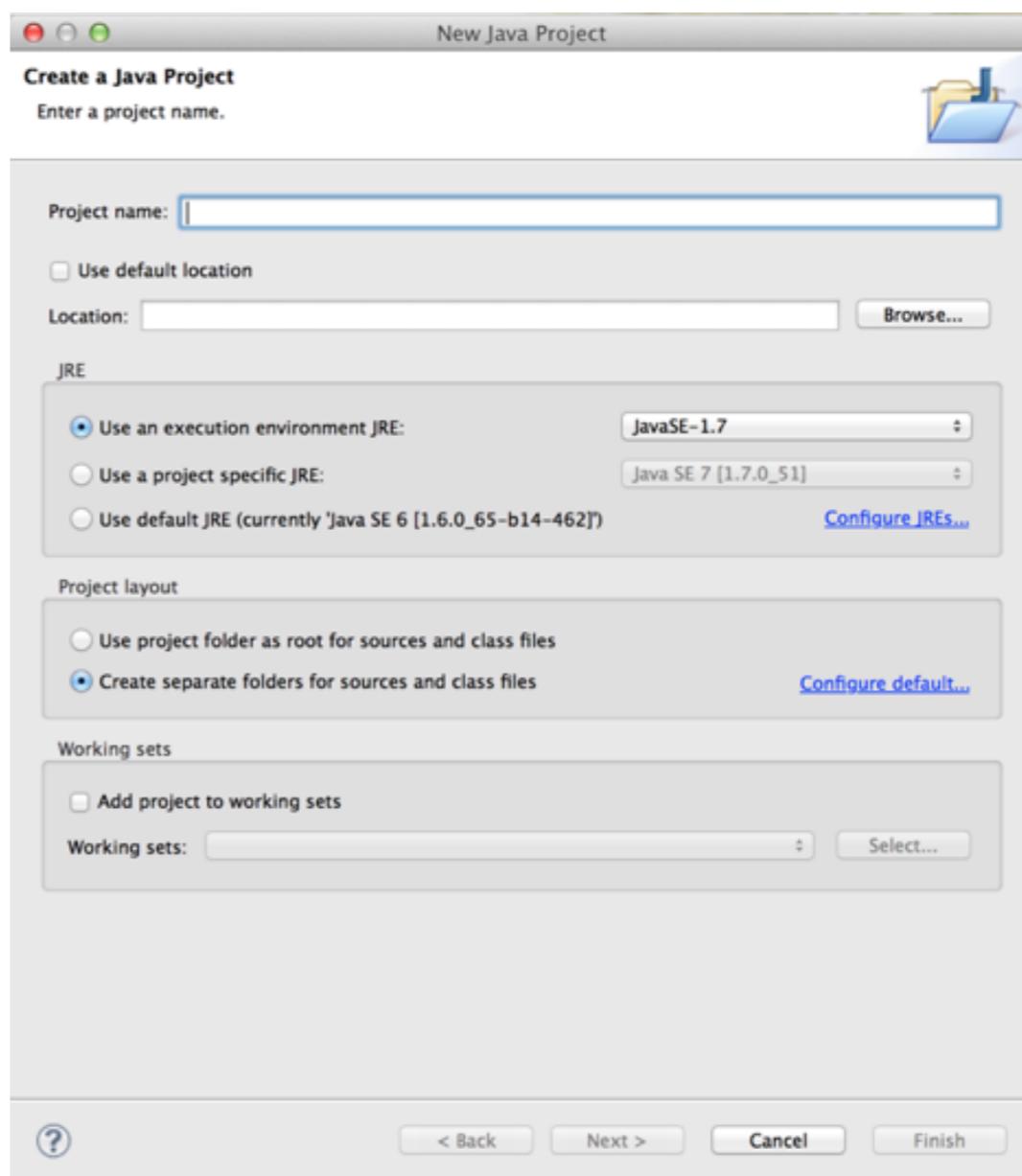
Setting up TestNG and WebDriver using Gradle

```
apply plugin: 'java'  
apply plugin: 'eclipse'  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    compile 'mysql:mysql-connector-java:5.1.29'  
    compile "org.testng:testng:6.8.7"  
    compile("org.uncommons:reportng:1.1.2") {  
        exclude group: "org.testng", module: "testng"  
    }  
    compile "com.jcraft:jsch:0.1.44-1"  
    compile "org.seleniumhq.selenium:selenium-java:2.40.0"  
    compile "javax.servlet:servlet-api:2.5"  
    compile "net.sourceforge.jexcelapi:jxl:2.6.12"  
}  
  
version = '1.0'  
  
jar {  
    manifest.attributes provider: 'gradle'  
}  
  
task run(type: JavaExec, dependsOn: classes) {  
    main = 'org.testng.TestNG'  
    args = ['build/resources/main/testng.xml'].toList()  
    classpath configurations.runtime  
    classpath sourceSets.main.output  
}
```

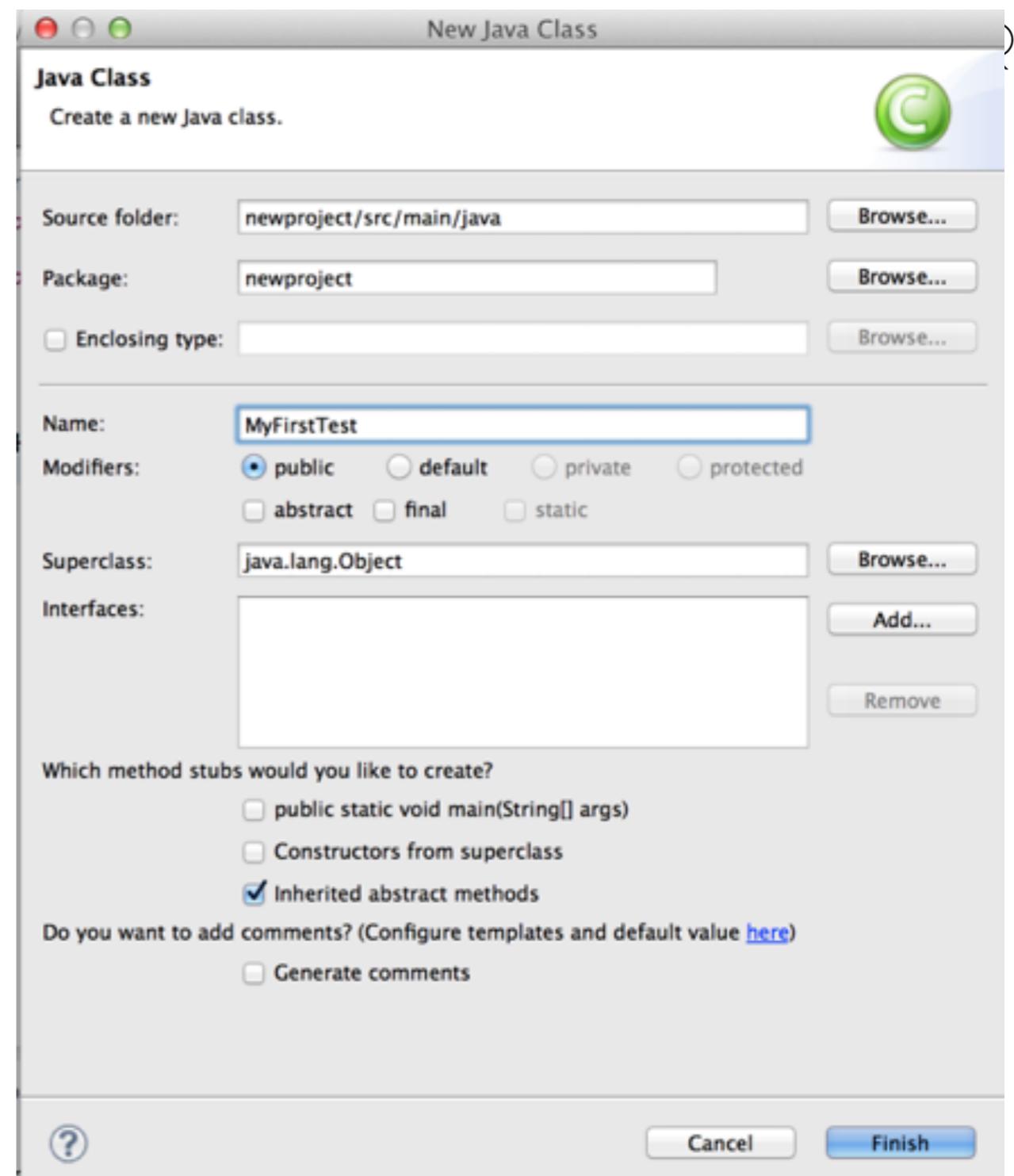
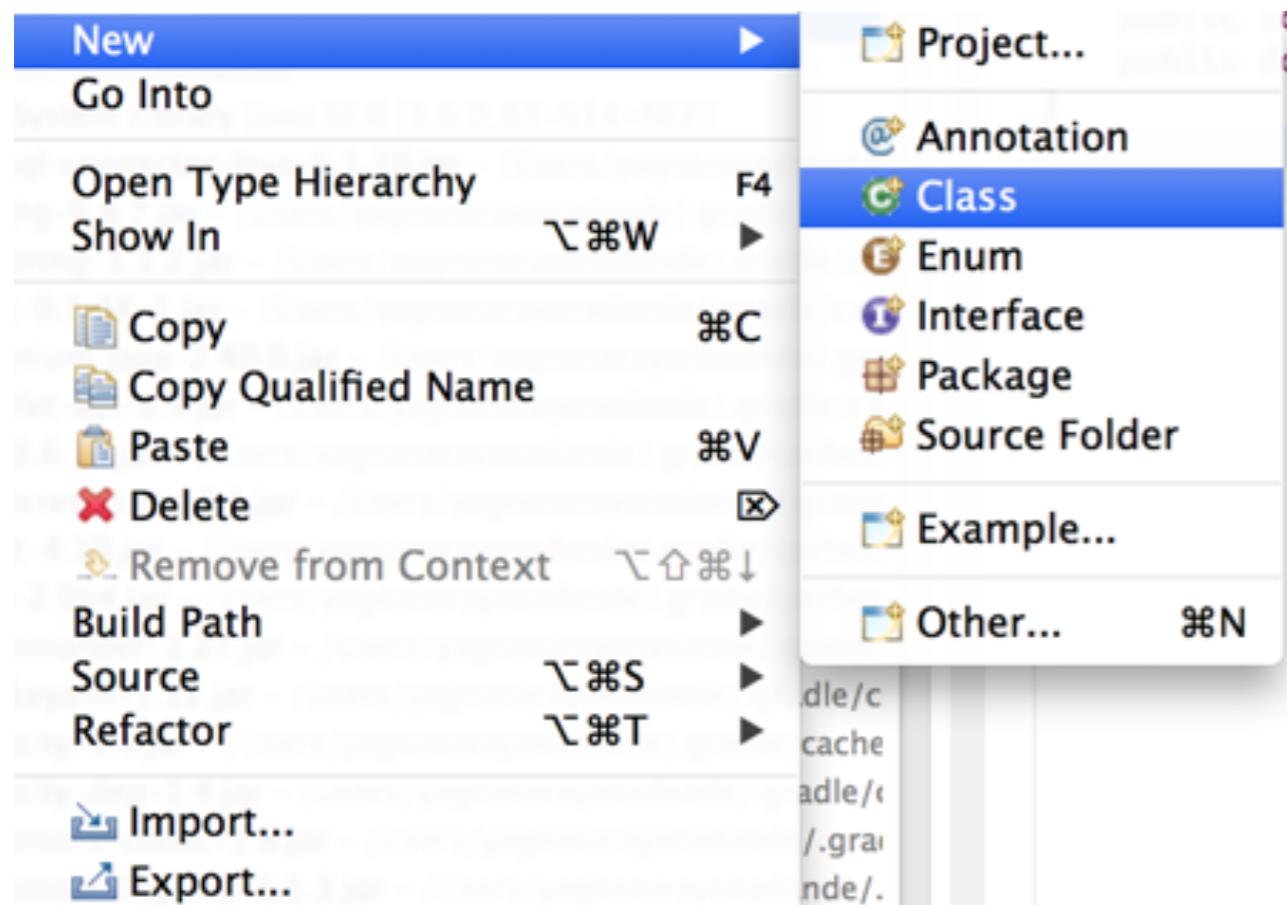
Create Project in Eclipse



Creating Project



Creating New Class



Writing First Test

```
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;

public class MyFirstTest {

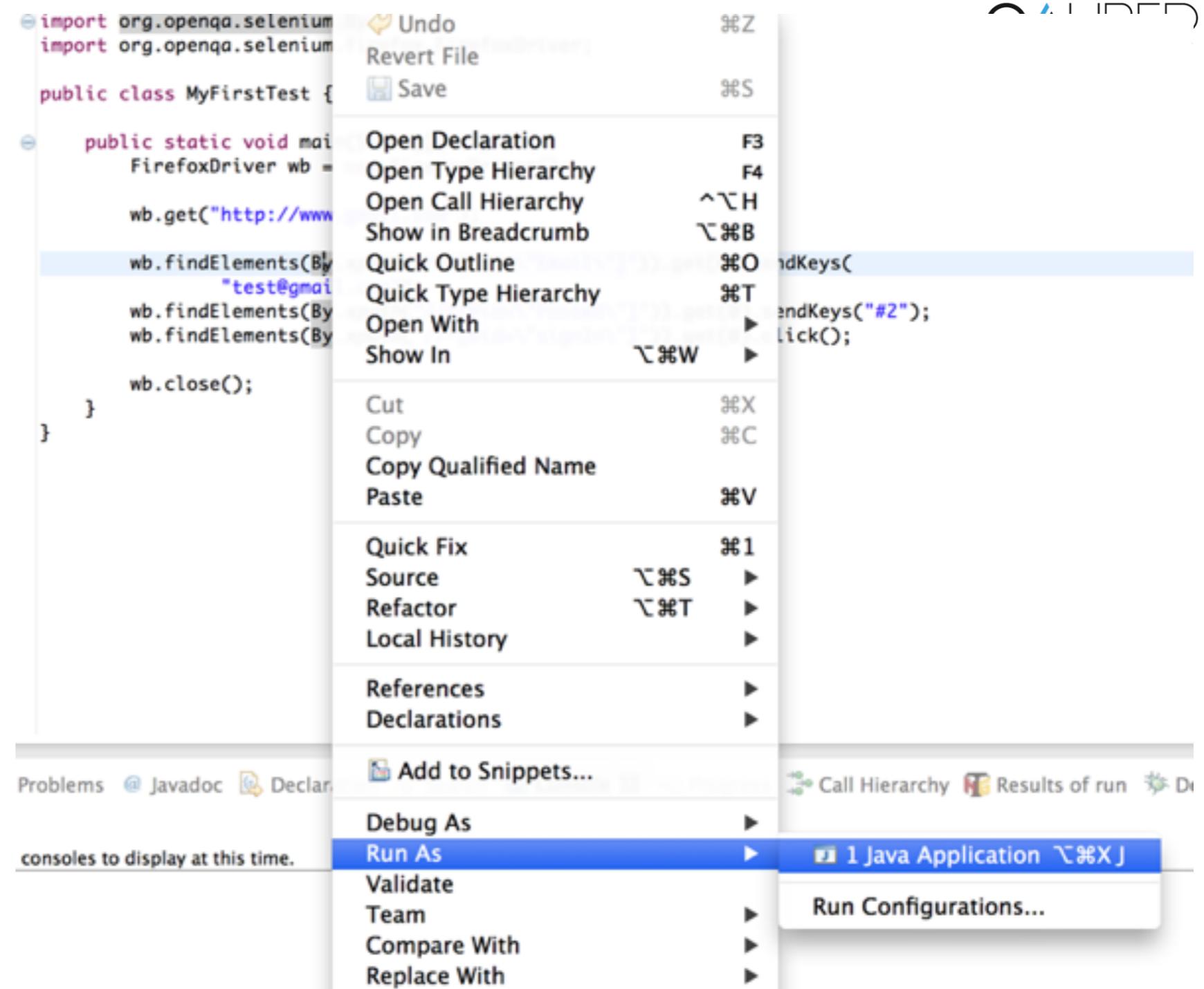
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "/Users/ydande/Desktop/chromedriver");
        ChromeDriver wb = new ChromeDriver();
        wb.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);

        wb.get("http://www.gmail.com");

        wb.findElements(By.xpath("//*[@id='Email']")).get(0).sendKeys(
            "yagna.bitspilani@gmail.com");
        wb.findElement(By.id("next")).click();
        wb.findElements(By.xpath("//*[@id='Passwd']")).get(0).sendKeys("#2");
        wb.findElements(By.xpath("//*[@id='signIn']")).get(0).click();

        wb.close();
    }
}
```

Running Test



Introduction to Java



1159 Sonora Court Suite 205,
Sunnyvale CA, 94086 USA Tel. +1
(408) 220 9912 Ext 1
www.qauber.com

Java

Java programming language was originally developed by Sun Microsystems.

Java is guaranteed to be Write Once, Run Anywhere.

Java

Object Oriented – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent – Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple – Java is designed to be easy to learn. If you understand the basic concept of OOP, it would be easy to master.

Secure – With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Java

Robust – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded – With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

High Performance – With the use of Just-In-Time compilers, Java enables high performance.

Dynamic – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

JVM

JVM is short for Java Virtual Machine.

It is a platform-independent execution environment that converts Java bytecode into machine language and executes it.

Most programming languages compile source code directly into machine code that is designed to run on a specific microprocessor architecture or operating system, such as Windows or UNIX.

Java is compile once and run anywhere language.

OOP

The object-oriented is a programming paradigm where the program logic and data are weaved.

OOP languages permit higher level of abstraction for solving real-life problems. The traditional procedural language (such as C and Pascal) forces you to think in terms of the structure of the computer (e.g. memory bits and bytes, array, decision, loop) rather than thinking in terms of the problem you are trying to solve. The OOP languages (such as Java, C++ and C#) let you think in the problem space, and use software objects to represent and abstract entities of the problem space to solve the problem.

Classes and Objects

What is an Object?

Any entity is an Object either in real world or in programming world

Then what is Class?

Blue print of a particular type of Object

Program Structure

```
public class BankAccount
{
    private int AccountNumber;
    private boolean Active;
    private double Balance;

    public BankAccount(int AccountNumber, double Balance, boolean Active)
    {
        this.AccountNumber = AccountNumber;
        this.Balance = Balance;
        this.Active = Active;
    }

    public void Withdraw(double amount)
    {
        Balance = Balance - amount;
    }

    public void Deposit(double amount)
    {
        Balance = Balance + amount;
    }

    public void getBalance()
    {
        System.out.println("Balance is "+Balance);
    }
}

public class Test
{
    public static void main(String args[])
    {
        BankAccount myAccount = new BankAccount(12345, 125.50, true);
        myAccount.Deposit(500);
        myAccount.getBalance();
    }
}
```

Essentials of Object

What is method?

Some form of Work - Behavior

Signature of Method - Method name and parameters. But not return value.

What are attributes?

Values which will determine state and Identity of an Object

Characteristics of Object

Identity

Responsibility - Use

State

Behavior - operations

Example:

Object: "Bank Account"

Identity: Account Number

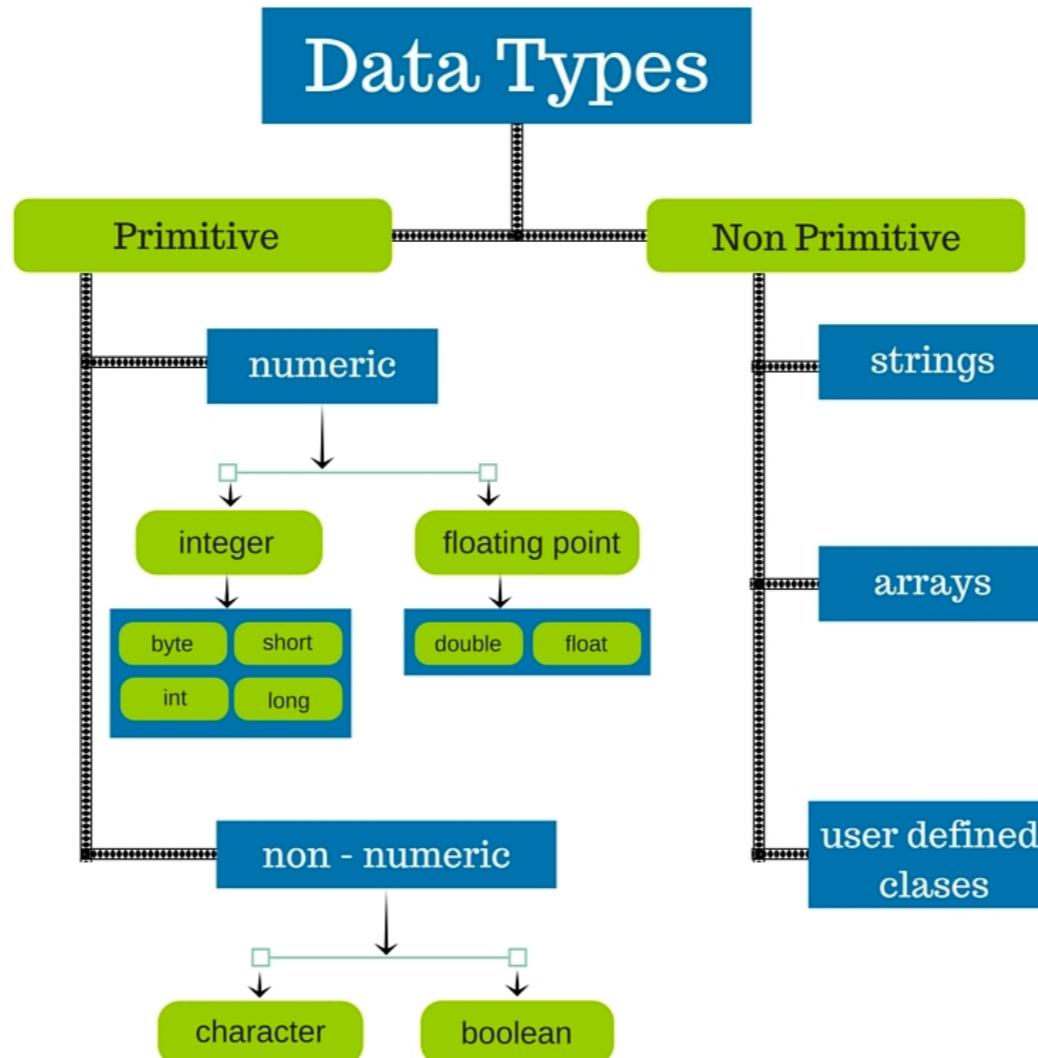
Responsibility: Save Money

State: Active, Balance

Behavior: Withdraw, Deposit

Basic data types

int(4 byte)
long(8 byte)
float(4 byte)
double(8 byte)
boolean(1 bit)
char(2 byte)



Variable types

Access Specifiers:

Types:

Class Access Specifier	- public, default
Attribute Access Specifier	- public, protected, default, private
Method Access Specifier	- public, protected, default, private
<hr/>	
where	public
	protected
	default
	private
<hr/>	

Decision making

If else Condition

```
public void Withdraw(double amount)
{
    if(amount<Balance)
    {
        Balance=Balance-amount;
    }else{
        System.out.println("You have insufficient Balance ...");
    }
}
```

Arrays & Loops

```
public class Student
{
    private String Name;
    private int RollNumber;

    public Student(String Name, int RollNumber)
    {
        this.Name = Name;
        this.RollNumber = RollNumber;
    }
    public String getName()
    {
        return Name;
    }
    public int getRollNumber()
    {
        return RollNumber;
    }
}
public class Driver
{
    public static void main(String[] args)
    {
        Student[] S = {new Student("s1",1), new Student("s2",2), new Student("s3",3)};
        printStudentDetails(S);
    }
    public static void printStudentDetails(Student[] S)
    {
        for (int i=0;i<S.length;i++)
        {
            System.out.println("Student Name: "+S[i].getName());
            System.out.println("Student Roll: "+S[i].getRollNumber());
        }
    }
}
```

String

String is an Object.

String Object can be created without using new keyword

String Class has following methods

equals

replace

subString

toLowerCase()

toUpperCase()