**Install Maven in Desktop**   
  
Follow instructions below to setup maven in the desktop: 

1. Download latest version of maven from internet or get the installation zip file from trainer
2. Extract the zip file into D: drive
3. Set the windows environment variables JAVA\_HOME, MAVEN\_HOME
4. Include PATH with MAVEN\_HOME\bin
5. Test maven setup with ‘mvn -version’ command in command line

**Create web project using Maven**   
  
path=

1. Open command line
2. Change the folder to the root folder of the Eclipse workspace
3. Execute the following command to create the maven web project structure. Change the proxyUser and proxyPassword accordingly in the command below.

mvn archetype:generate -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456 -Dhttp.proxyPassword=password -DgroupId=com.cognizant.maven -DartifactId=mvnlearn -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false

1. Build the project using the following command:

For changing move to the project directory

   mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050

1. Switch to eclipse and select File->import->projects from folder

**SME to explain how maven works** 

1. Java libraries are available in central maven repository in internet ([https://repo.maven.apache.org](https://repo.maven.apache.org/))
2. Maven downloads dependent jar files from the central maven repository and places them in C:\Users\[EMP\_ID]\.m2\repository folder. This is the local maven repository.
3. This transfer is a one-time activity and download happens only when new dependencies are added in pom.xml.

Refer diagram in Maven Repositories topic of <http://tutorials.jenkov.com/maven/maven-tutorial.html> for understanding the Maven Architecture.

**Setup Maven project in Eclipse**   
  
Follow steps below to setup the Maven project in Eclipse. 

1. Import the maven project folder in previous hands on in Eclipse (File > Import > Maven > Existing Maven Projects and select the new maven project folder)
2. Make following change in pom.xml
   1. Change packaging as war <packaging>war</packaging >
   2. Change compiler version from 1.7 to 1.8
3. Add Tomcat Server Environment as Library (Right Click Project > Build Path > Configure Build Path > Add Library > Server Runtime > Apache Tomcat v9.0)
4. Add the project to Tomcat Server by right clicking the Tomcat Server in Servers perspective and use "Add and Remove" option.
5. Start the server and check if the application is running.

**Create a Spring Web Project using Maven**   
  
Follow steps below to create a project: 

1. Go to <https://start.spring.io/>
2. Change Group as “com.cognizant”
3. Change Artifact Id as “spring-learn”
4. Select Spring Boot DevTools and Spring Web
5. Create and download the project as zip
6. Extract the zip in root folder to Eclipse Workspace
7. Build the project using ‘mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456’ command in command line
8. Import the project in Eclipse "File > Import > Maven > Existing Maven Projects > Click Browse and select extracted folder > Finish"
9. Include logs to verify if main() method of SpringLearnApplication.
10. Run the SpringLearnApplication class.

SME to walk through the following aspects related to the project created:

1. src/main/java - Folder with application code
2. src/main/resources - Folder for application configuration
3. src/test/java - Folder with code for testing the application
4. SpringLearnApplication.java - Walkthrough the main() method.
5. Purpose of @SpringBootApplication annotation
6. pom.xml
   1. Walkthrough all the configuration defined in XML file
   2. Open 'Dependency Hierarchy' and show the dependency tree.