Name : Sakshi Mohan Kamble

PRN : 2020BTEIT00205

Course : Agile software tools and practices

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 13**

**Title:**

**Use tool Maven for learning and demonstrating the (dependencies) SDLC (software-dev-lifecycle) for java codes/projects**

Maven helps to automate the build process by defining a standard build lifecycle, configuring project dependencies, and managing project dependencies.

Maven uses a configuration file called "pom.xml" to manage project dependencies and other configuration settings. The pom.xml file contains information such as the project name, description, version, dependencies, and plugins.

Maven follows a standard build lifecycle consisting of different phases such as compile, test, package, install, and deploy. Each phase is associated with a set of goals or tasks that need to be executed.

Maven uses a central repository called Maven Central to store project dependencies. When a project needs a specific dependency, Maven downloads it from the central repository and includes it in the project.

Maven also supports plugins that can be used to extend its functionality. There are numerous plugins available for Maven that can be used to perform various tasks such as code analysis, documentation generation, and deployment.

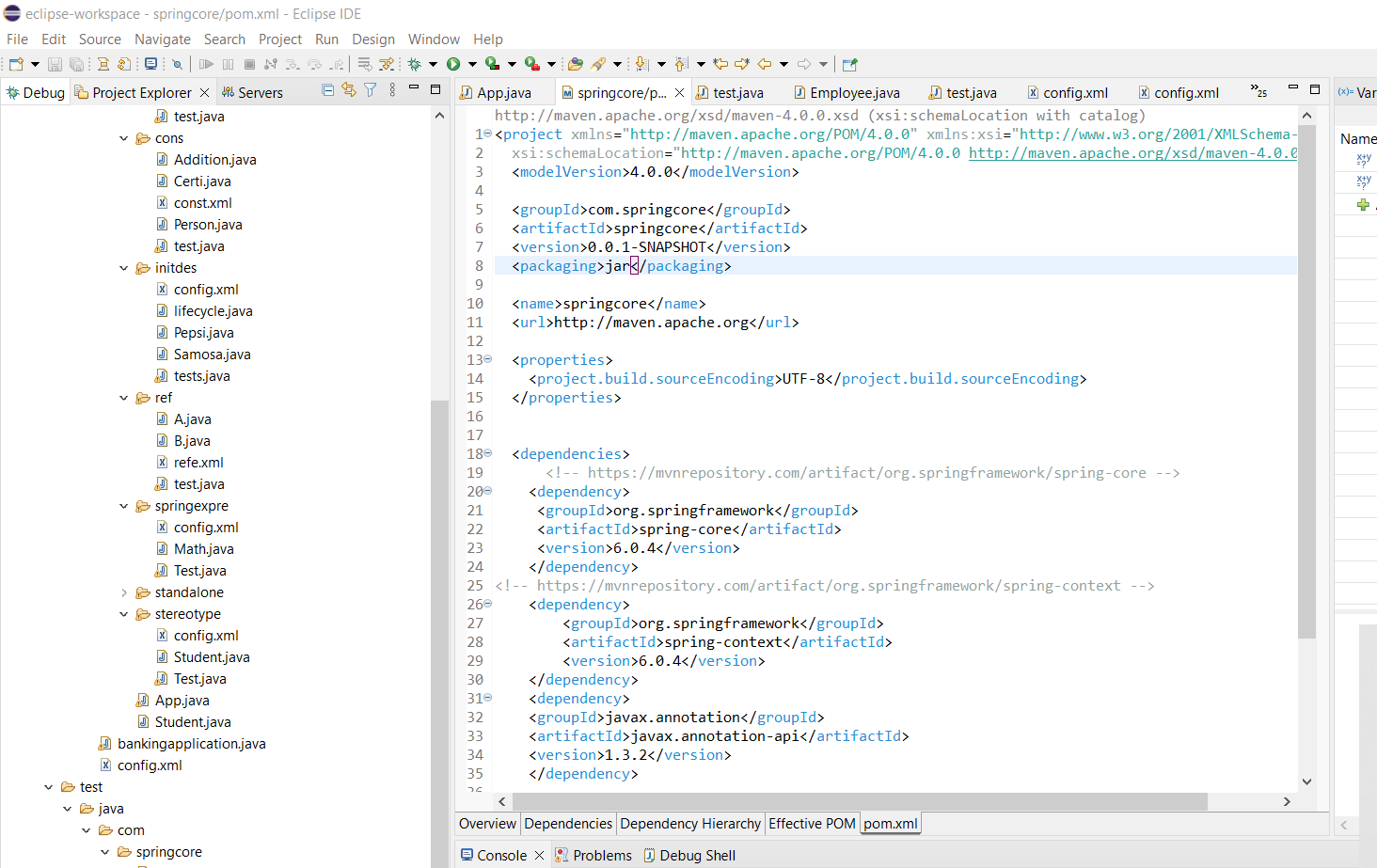
Overall, Maven is a powerful tool that helps simplify the build process for Java projects and makes it easier to manage project dependencies.

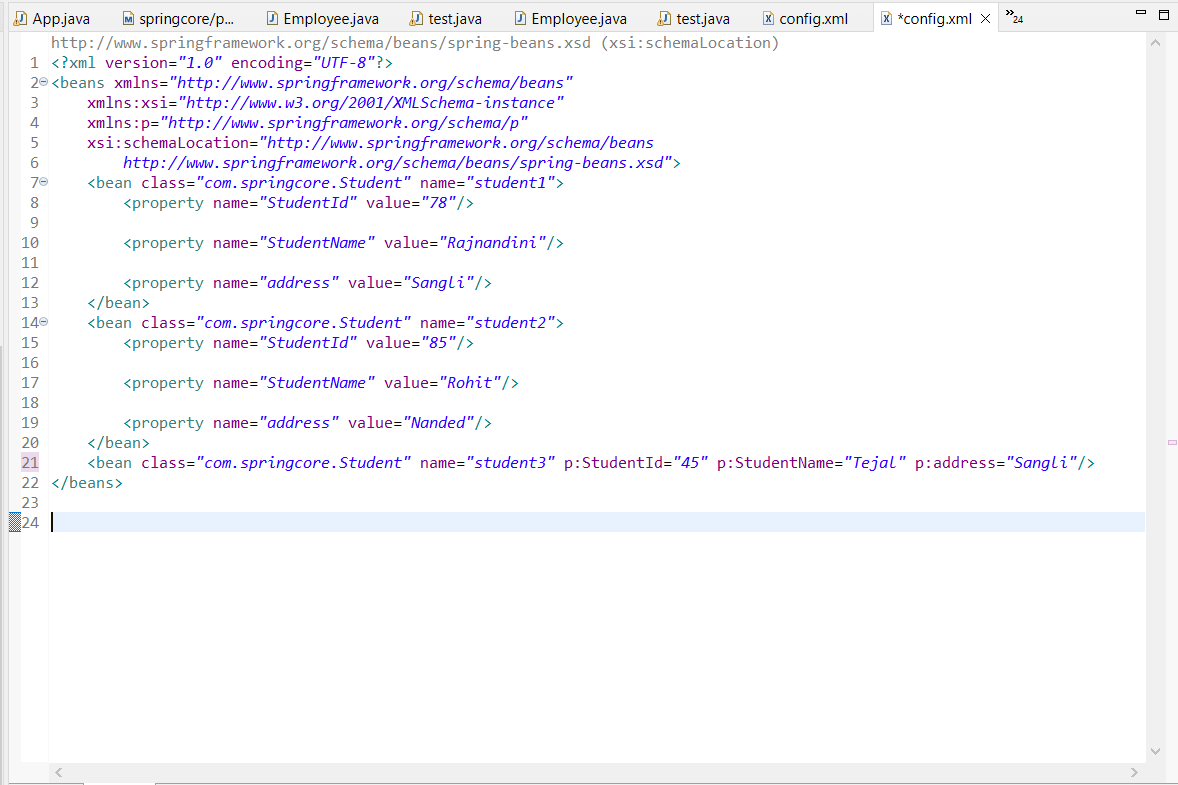
Need for Maven:

Maven helps developers manage Java-based apps through projects that organize and handle code files & build scripts to execute and run compiler tools, version no. for compiled code, plus dependency management that allows one project to reference a version of other projects.

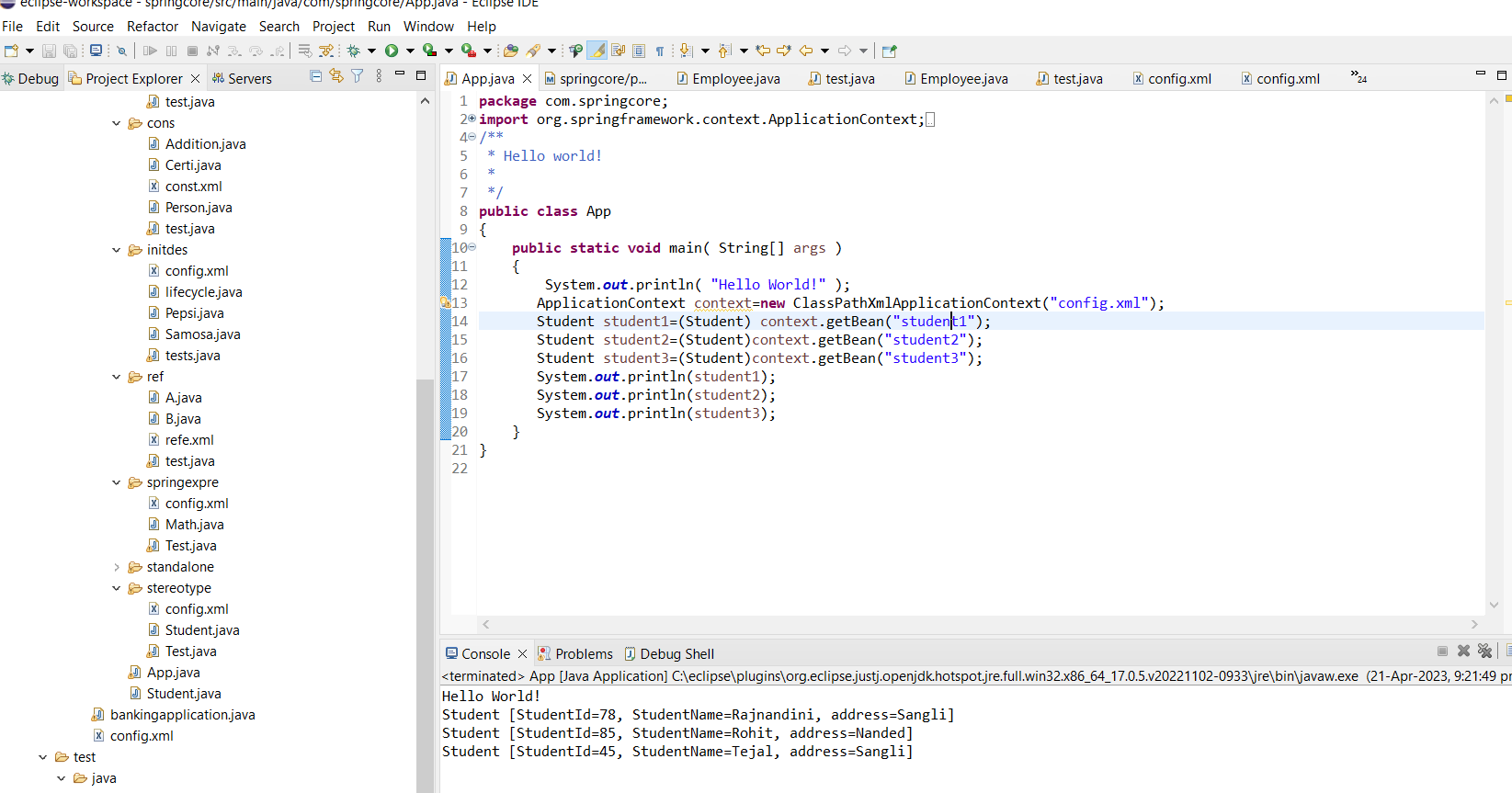
* Build a project using Maven.
* Accumulation and collection of Source Code
* Running Tests (functional tests and unit tests)
* Upload the packages to remote repos (Artifactory, Nexus)
* Packaging the outcomes into WAR’s, JAR’s, RPM’s, etc.
* Using Maven you can add jars & other dependencies of the project effortlessly.
* Maven gives project information (dependency list, unit test reports, log document, etc.)
* Using Maven we can simply integrate our project with a source control system (like Git or Subversion).

Screenshots:

**POM.xml****config.xml**



**App.java**



References:

* [https://www.browserstack.com/guide/maven-devops#:~:text=Maven%20provides%20software%20developers%20build,%2C%20Python%2C%20Ruby%2C%20etc.](https://www.browserstack.com/guide/maven-devops%23:~:text=Maven%20provides%20software%20developers%20build,%2C%20Python%2C%20Ruby%2C%20etc.)
* <https://maven.apache.org/>