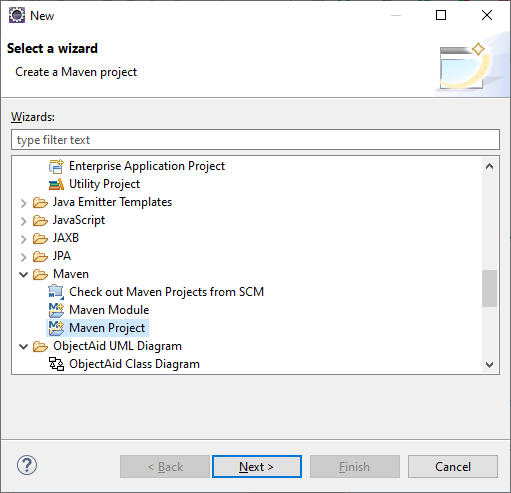
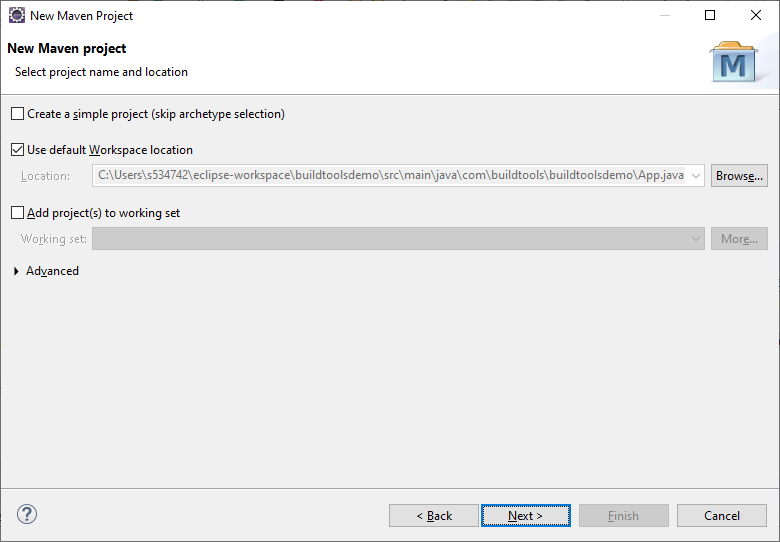
**CT: App Design: Patterns and Frameworks**

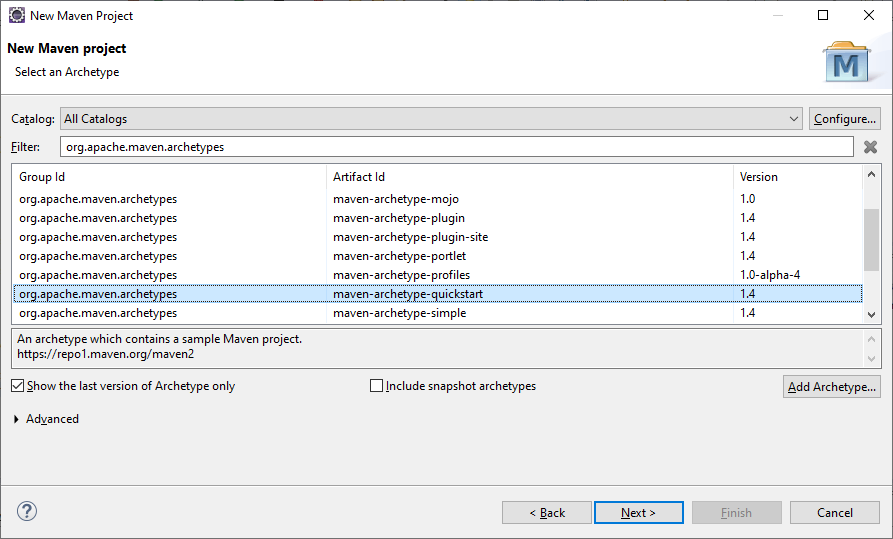
**Creating Maven Project:**

1. Choose File > New > Other to open the New Project wizard
2. Select Maven and then choose Maven Project as shown below and click Next

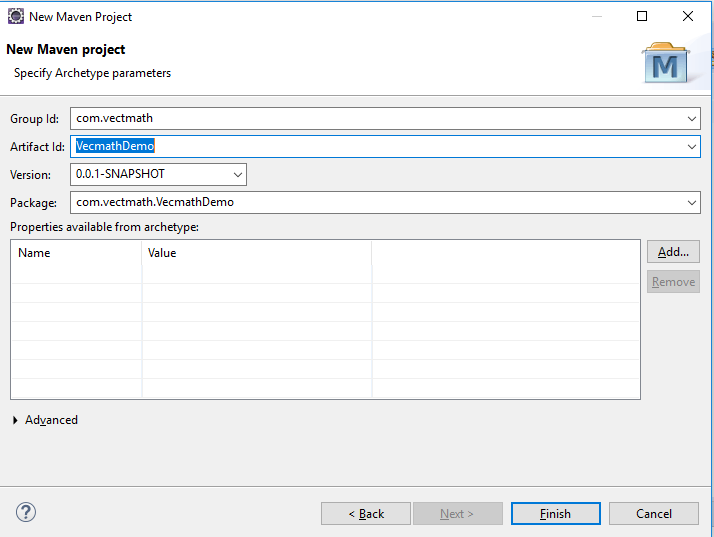




1. Leave everything as default and click Next
2. For this project we will use **Artifact ID**: maven-archetype-quickstart. To get this type “org.apache.maven.archetypes” in Filter and click Next



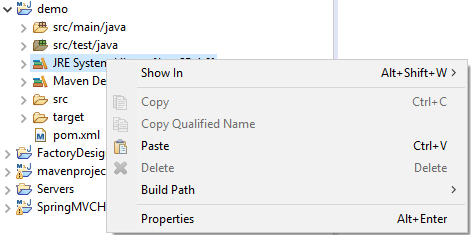
1. Now you need to enter **Group Id** and **Artifact Id.**
2. For e.g. **Group Id:** com.vectmath
3. **Artifact Id:** VecmathDemo and click Finish.



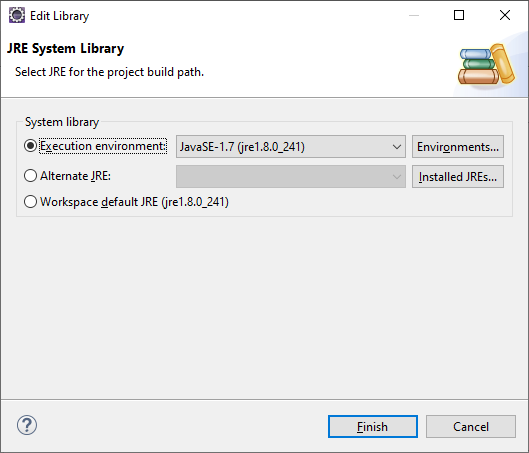
1. If it is asking to confirm anything in console, enter “Y” and click on enter button. This will create your project.

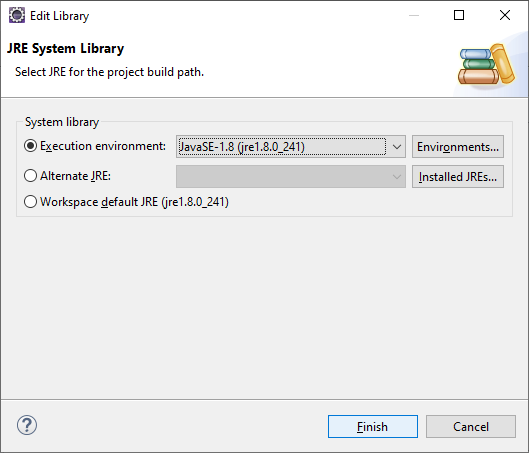
**Changing the Java version:**

1. Now you need to change JavaSE from 1.7 to 1.8
2. In project, right click on the **JRE System Library** and click on properties.



1. Now change JavaSE-1.7 to JavaSE-1.8 from the dropdown under **Execution Environment** and click Apply and close.





1. In App.java we are doing matrix operations using Matrix3d
2. To read matrix from the keyboard add the below mentioned code blocks to App.java

**private** **static** Matrix3d readMatrix() {

// Create a scanner object.

Scanner scanner = **new** Scanner(System.***in***);

// Read input elements for matrix from user.

**double**[] matrixElements = **new** **double**[3 \* 3];

**for** (**int** index = 0; index < matrixElements.length; index++)

{

matrixElements[index] = scanner.nextDouble();

}

// Create and return the matrix.

**return** **new** Matrix3d(matrixElements);

}

**public** **static** **void** main(String[] args) {

// Declare two matrix variables

Matrix3d matrix1, matrix2;

// Read input for matrix1

System.***out***.println("Enter input elements for 3X3 matrix#1: ");

matrix1 = *readMatrix*();

// Read input for matrix2

System.***out***.println("Enter input elements for 3X3 matrix#2: ");

matrix2 = *readMatrix*();

Matrix3d resultMatrix = **new** Matrix3d();

// Matrix addition

System.***out***.println("The addition of two matrices is:");

resultMatrix.add(matrix1, matrix2);

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

// Matrix multiplication

System.***out***.println("The multiplication of two matrices is:");

resultMatrix.mul(matrix1, matrix2);

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

}

**Adding dependency for vecmath in pom.xml**

1. Open pom.xml file and add the following code inside the < dependencies> tag. Once you add the dependency to file, build the project.
2. **Note**: If your file does not contain the tag, add < dependencies> </ dependencies> below the <packaging> tag and above the <properties> tag.

<dependency>

<groupId>javax.vecmath</groupId>

<artifactId>vecmath</artifactId>

<version>1.5.2</version>

</dependency>

1. Make necessary imports and run the application.