**SOFTWARE ENGINEERING DEPARTMENT**

**Course: CMPE 202** – SOFTWARE SYSMTES ENGINEERING

**Student name/ID:** HARISHKUMAR KADIROMPALLIVENKATASHIVAREDDY (010712013)

**Personal Project:** UML PARSER

**Semester:** FALL 2015

**UML Parser**

Goal:

Develop a utility to parse Java Source Code and create UML Class Diagram.

Design:

This utility is designed using 3rd party libraries like Java Parser and Plant UML, parser will convert Java Source File into Abstract Syntax Tree, and syntax tree will be accepted by UML Parser utility and convert tree notation into the language syntax understood by UML generator tool. In this project I have used Plant UML as UML generator tool.

System Configuration:

1. Install JRE 1.8 version. Some part of the code is having syntax newly introduced in Java 1.8. If you don’t have JRE 1.8 version configured in your system download it from the link shown below:-
2. Plant UML requires UML GraphViz Dot software to be installed in your system to generate diagrams. Please follow the instructions as said below to download and configure GraphViz dot software:-
   1. Download the software from the link shown below :-
      1. For Windows

<http://www.graphviz.org/Download_windows.php>

* + 1. For Mac

<http://www.graphviz.org/Download_macos.php>

* 1. Recommended to Install in the default directory
     1. Windows
        1. Firstly check in: C:\Program Files\Graphviz\*\bin\dot.exe
        2. If not found in previous path, Then check in: C:\Program Files (x86)\Graphviz\*\bin\dot.exe
     2. On Linux/Mac OS-X
        1. First check in: /usr/local/bin/dot
        2. Then in /usr/bin/dot

Above installation instructions should configure Graphviz properly in your machine. In any case if there are any issues please refer to the link <http://plantuml.com/graphvizdot.html>

How to run:

Once you have configured your system, please follow the steps shown below to run the UML Parser Utility:

1. Open Command Prompt in Windows/Shell in Mac.
2. Locate the UML Parser Jar (Go to the directory in which UML Parser jar is downloaded)
3. Run the command as shown below:

Java –jar umlparser.jar “<input-directory>” “<output-filename>”

**<input-directory>** : Directory that contains Java Source Files. It won’t work for sub directories.

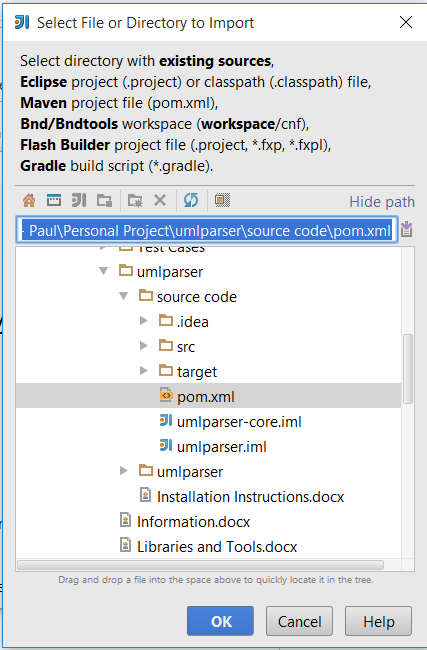
**<output-filename>** : Name of the class diagram image file. Format can be of “filename” OR “filename.png”. No other formats are supported.

How to compile and Jar Generation:

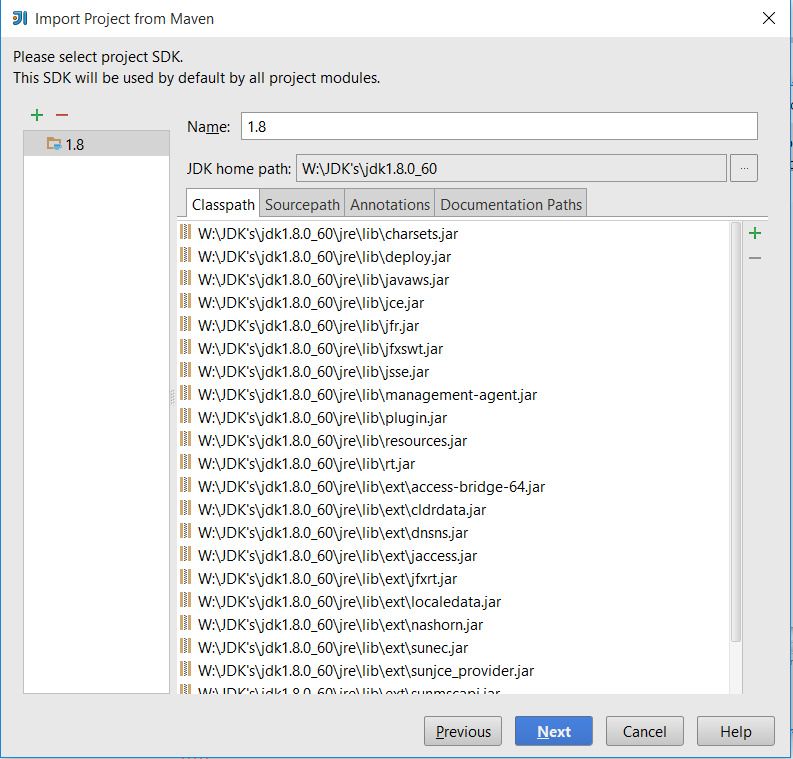
1. Open IntelliJ if it’s already installed in the machine. If not then download intellij from <https://www.jetbrains.com/idea/download/>
2. Once IntelliJ is downloaded and installed, open the IDE and click on “Import Project”.



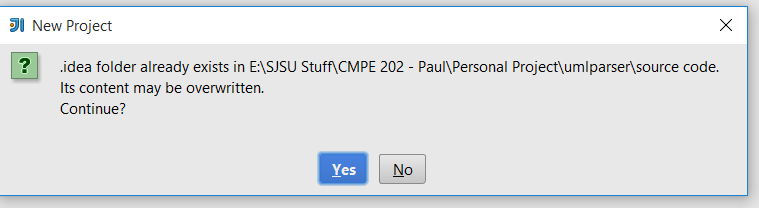
1. Select “pom.xml” file as shown from the source code folder



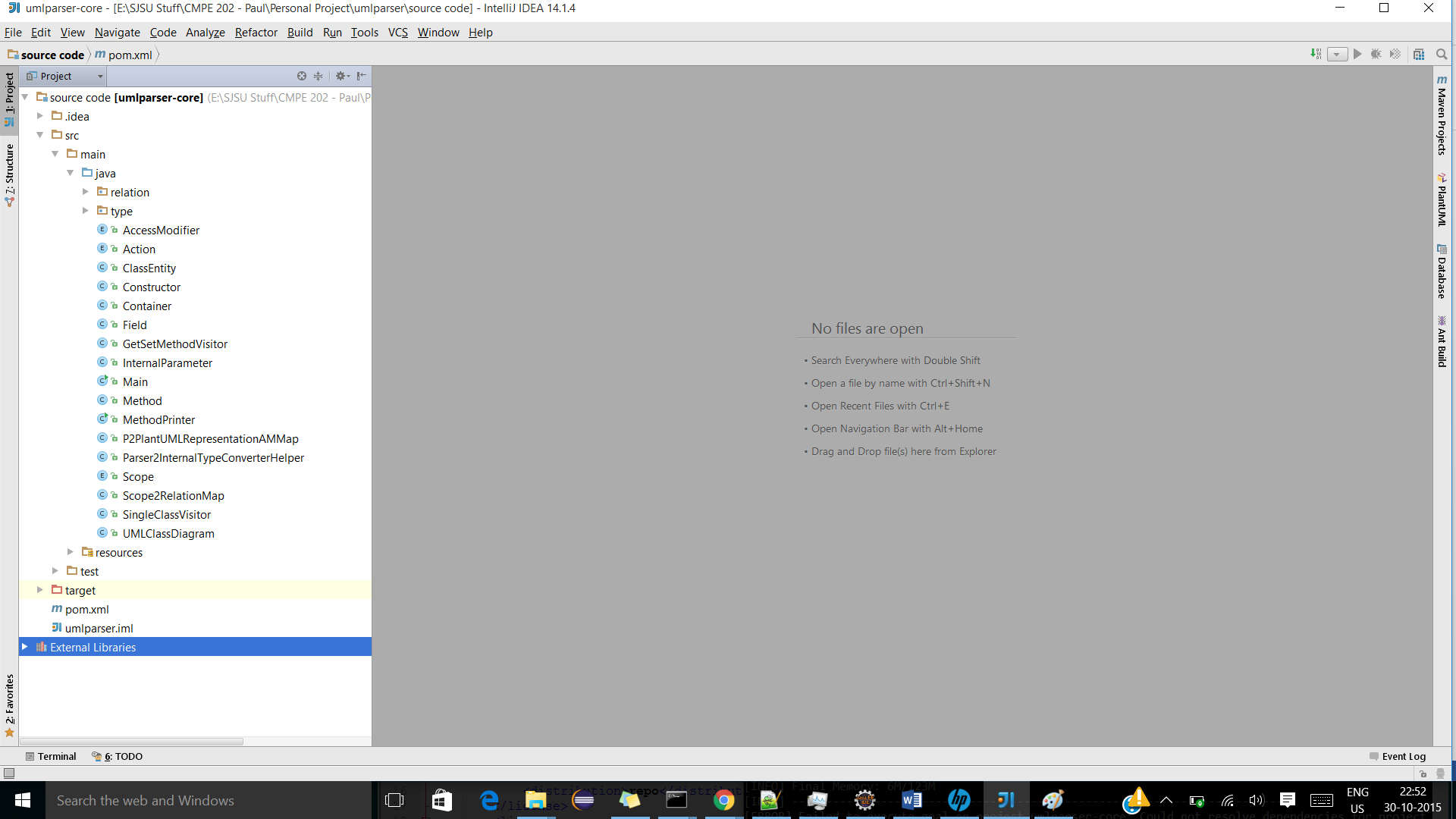
1. Click OK and Next until you reach the screen shown below and Ensure you have selected Java 1.8 version SDK.



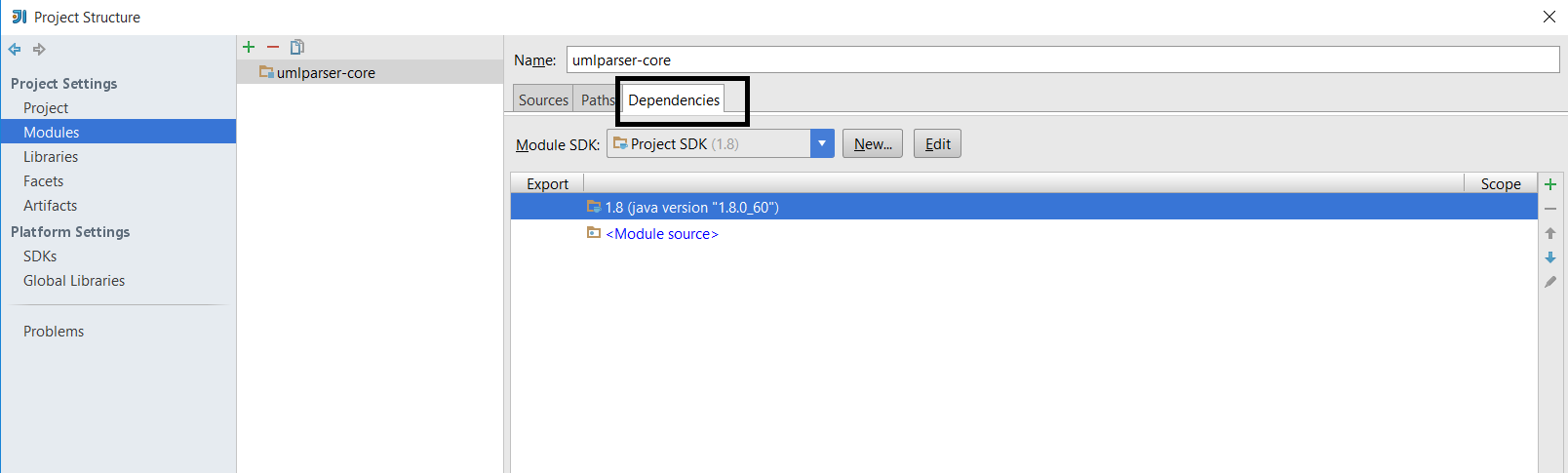
1. Click on NEXT and continue, until it shows the pop up as shown below and click “Yes” and continue



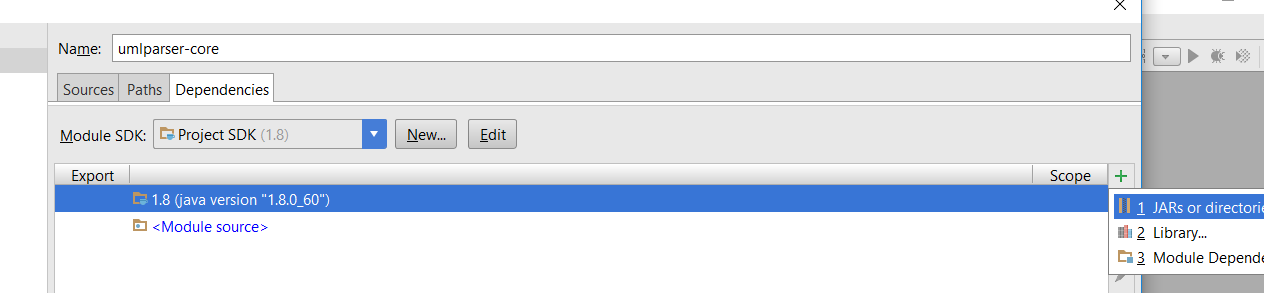
1. Now you should be able to see the screen as shown below:



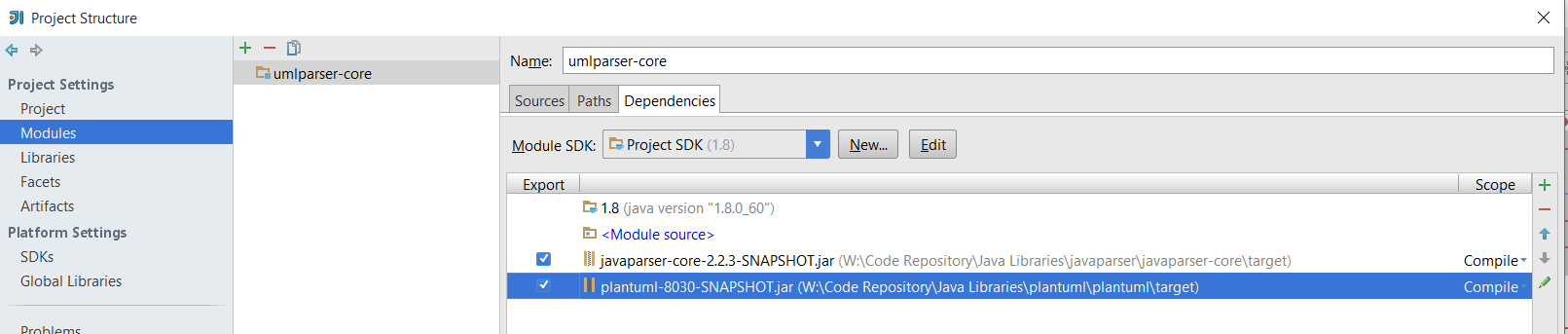
1. Now let us add the dependencies for my project.
2. Press “Ctrl+Alt+Shift+S” OR File->Project Structure and you will see a pop in which you have to select “Modules” and then Select dependencies tab as shown below.



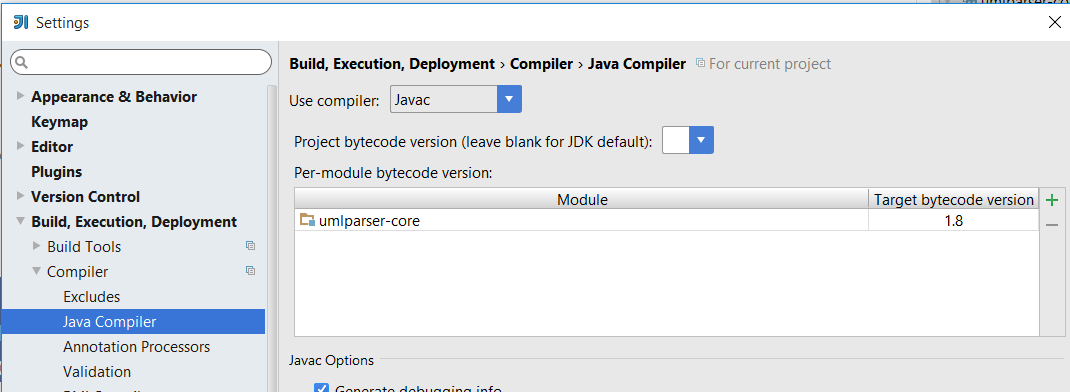
1. Click on the “+” button as shown in the image below and Select Jar’s OR Directories.



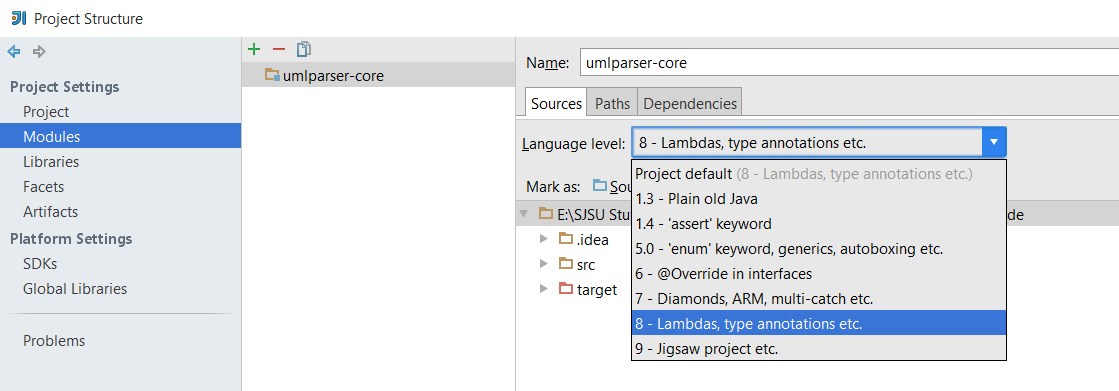
1. Select javaparser-core-2.2.3-SNAPSHOT.jar which is present in the dependencies folder and add it as a dependency.
2. Do the same procedure and also add plantuml.jar as a dependency to this project. Once you have done 10 and 11 your dependencies tab should look like this and make sure you have selected(Tick Mark) both the Jars.



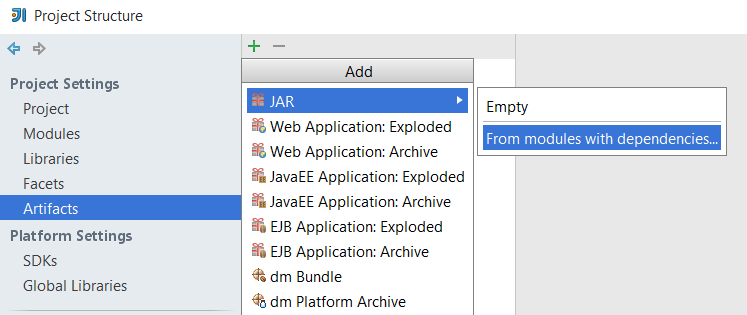
1. Once this is done click on Apply and dependencies are added into this project.
2. Before starting the build we have to make sure we have configured 1.8 SDK.
3. In order to check that please go to “Setting” – Ctrl + Alt+S , Build Execution Deployement -> Compiler -> Java Compiler and ensure you have selected 1.8 as shown below, if it is not 1.8 please change it to use 1.8 version :



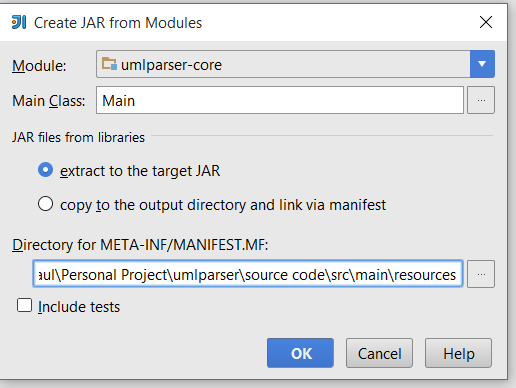
1. Lastly go to Project Settings – “Ctrl + Alt + Shift + S” and go to Modules , sources tab and then select “Language Level” as 8.0 and Click on OK. As shown below :



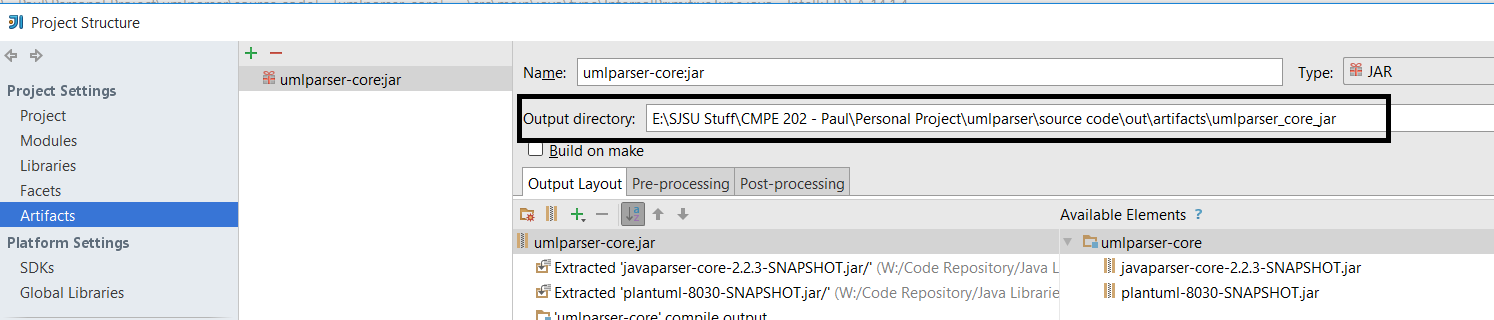
1. In order to generate a Jar for this project, again go to Project Structure Click on Artifacts and Select the options as shown below but before this please delete ManiFest file present under ~../src/main/resources:



1. Once you click on “from modules with dependencies” you will get a pop up to define main class in which give input as “Main” and also change the path of Mainfest file to be present under ~../src/main/resources folder instead of ~../src/main/java as shown below



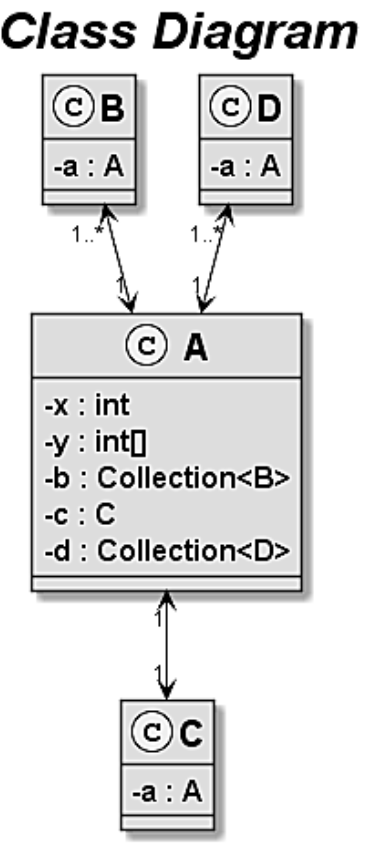
1. Click on OK, and note down the output path directory where Jar will be created as highlighted below and again Click on OK on Project Structure window.



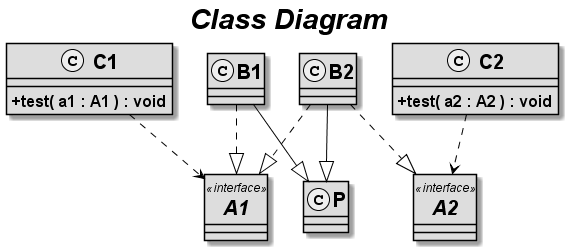
1. Now go to Build and Click on “Build Artifacts -> build” which will generate the Jar in the output directory highlighted above.
2. Now this Jar can be executed using the command shown in “How to Run” section above.

**OUTPUT SNAP SHOTS**

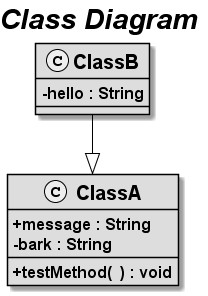
Test Case 1:



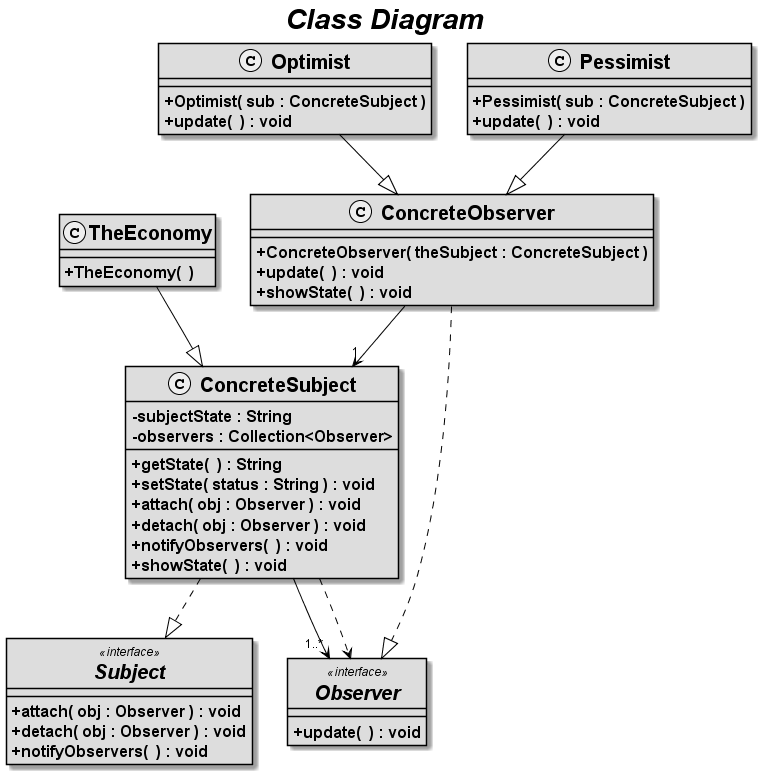
Test Case 2:



Test Case 3:



Test Case 4:



Test Case 5:

