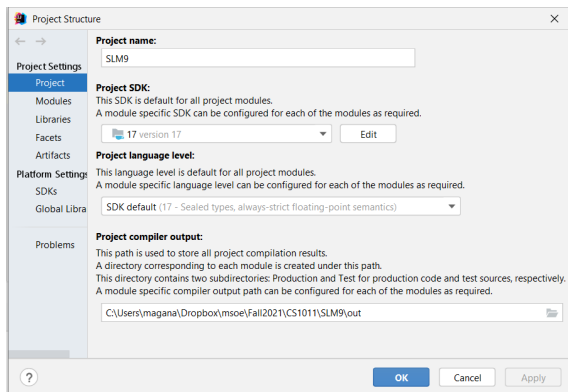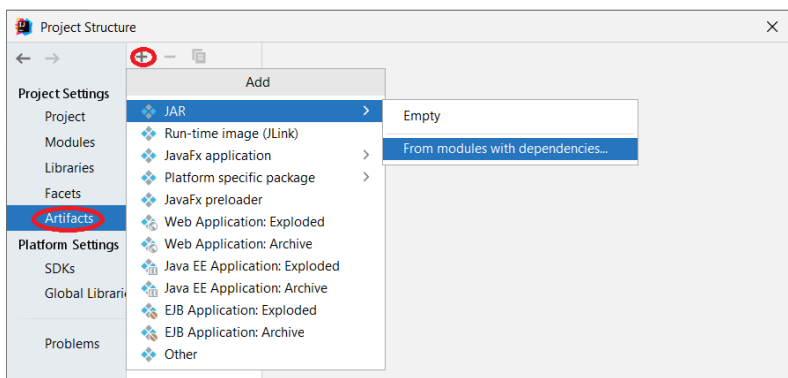SLM9 - Creating a JAR

Step 1: With a project open, navigate to the Project Structure (File->Project Structure)

Step 2: Select Artifacts and then click the plus (+), choose JAR and then select *From modules with dependencies*


Step 3: Choose the Main Class by clicking the folder icon on the Creating JAR From Modules window, then the class with the public static void main method. Here it is SLM9 (SLM9.java). Then click ok:
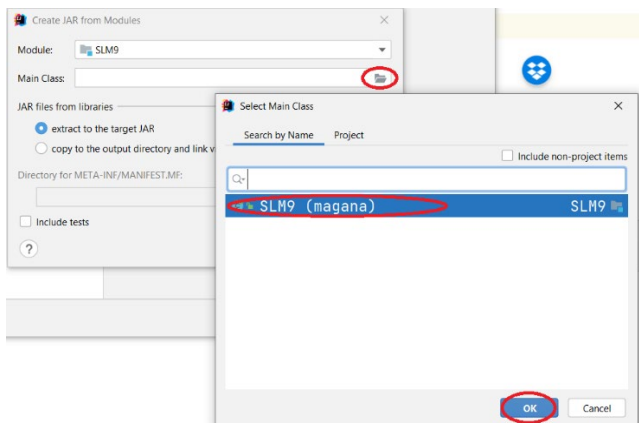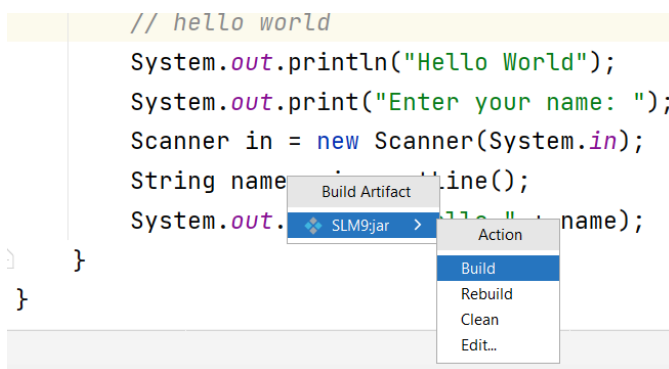
Step 4: Click ok on the following couple screens until you are back to your project. There will now be a MANAFEST.MF file inside of META-INF sub folder under src
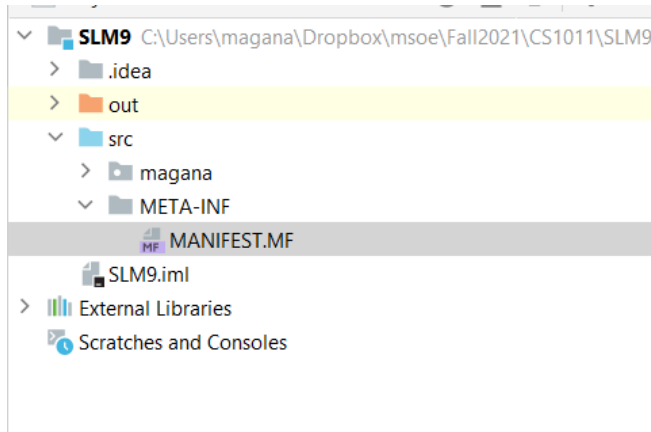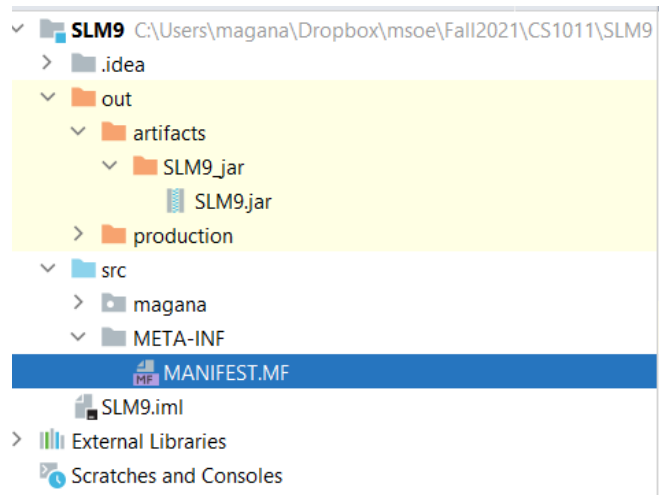

Step 5: Now we need to Build the JAR. To do this, you select Build->Build Artifacts
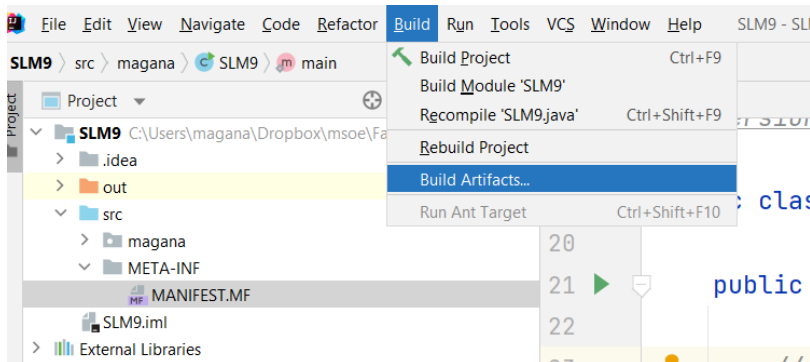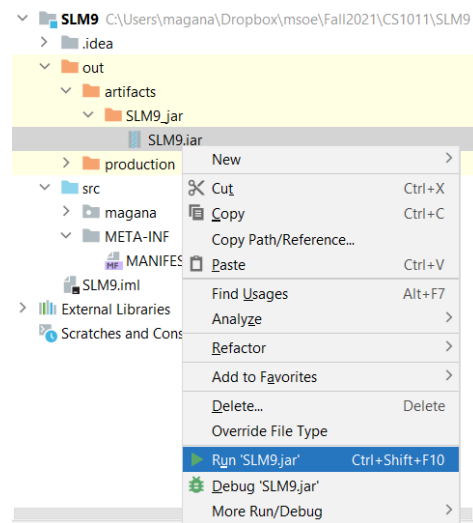

Step 6: And then select the Build action for our SLM9.jar
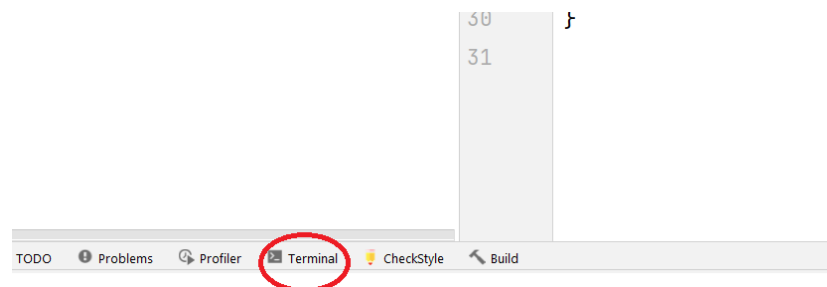
Step 7: The jar will now be in the …/out/artifacts folder. Note that the MANIFEST.INF contains information about the JAR, including the location main class (this is where the jar will start running).

Step 8: You can run the JAR by right clicking on it and choosing run:

Step 9: You can also run the jar from the command line (terminal) from within IntelliJ (click on Terminal):

Step 10: From within the Terminal screen, the command that works on my system is – BUT THIS WILL LIKELY NOT WORK FOR YOU:

**C:\java\jdk-17\bin**java -jar out/artifacts/SLM9_jar/SLM9.jar

```
Terminal:  Local ×   Local (2) ×   + ∨
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\magana\Dropbox\msoe\Fall2021\CS1011\SLM9> C:\java\jdk-17\bin\java -jar out/artifacts/SLM9_jar/SLM9.jar
Hello World
Enter your name: Jon
Hello Jon
PS C:\Users\magana\Dropbox\msoe\Fall2021\CS1011\SLM9> █
```

Step 11: Note that I had to specify the path to the java executable as "**C:\java\jdk-17\bin**java" because that is where I unzipped java17 when I originally installed IntelliJ. You may have extracted java elsewhere and would need to change that path to match. Here is how to find your path to the java executable:

To verify where you extracted java17, you can open the *Project Structure* again (see Step 1), and then click on SDKs along the left. Java 17 is selected in the center, and along the right we can see *JDK home path*. You can copy this path and then use it in Step 10 (don't forget to add "\bin\java" to the end of it):


C:\java\jdk-17 -> **C:\java\jdk-17\bin**java
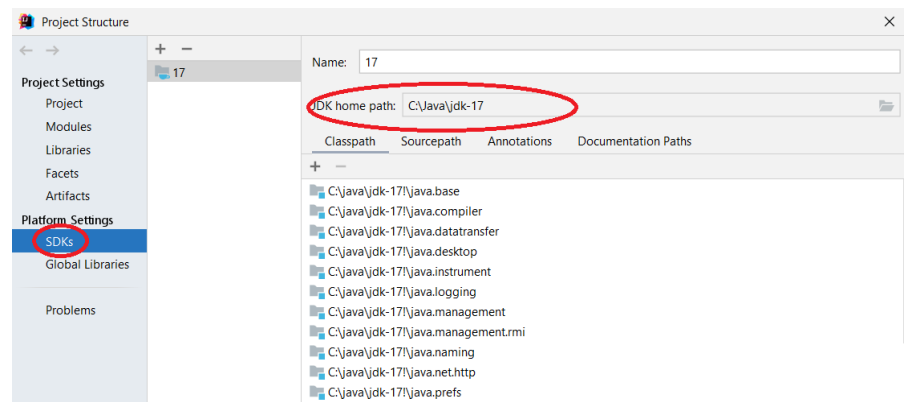
➔ Now that you have the path to java (java.exe) you can return to STEP 10 and use your path the run the program from the terminal within IntelliJ.

Step 12: If we navigate to that location from Step 11 (**C:\java\jdk-17** for me), and double click on the bin folder, we can see the java.exe executable:
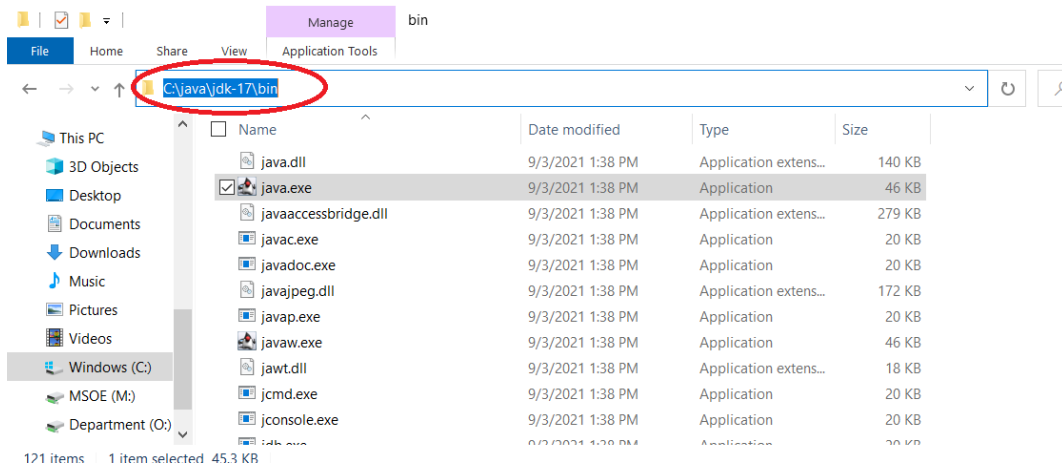
Step 12.1: Note that if you don't see file extensions, you can show file extensions by default in windows by clicking view, and then checking the "file name extensions" box:
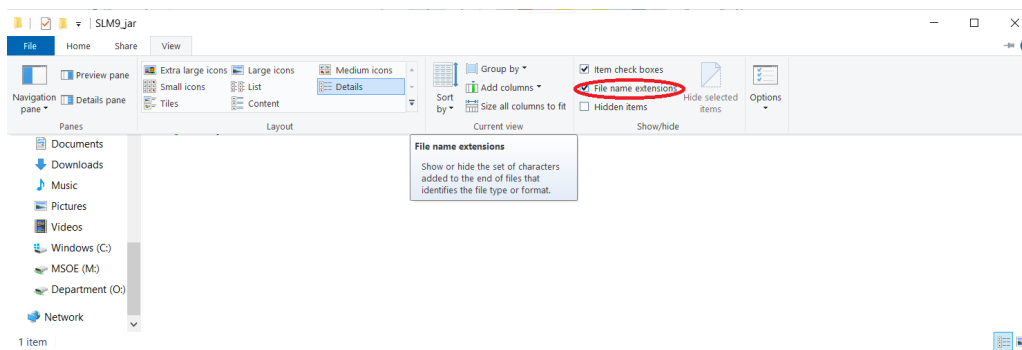

This concludes the first part of SLM 9 – follow your professor's instructions on submission.

A bonus section of the SLM follows on the next page.

Bonus: Running the jar using the windows command line

Step 1: Navigate to the folder in windows where the JAR is stored (from within your project, go to *out* folder and then *artifacts* folder). You should see the jar file itself. JARs use the zip file format, so you can old remain a something.jar to something.zip to be able to open it like a compressed folder:
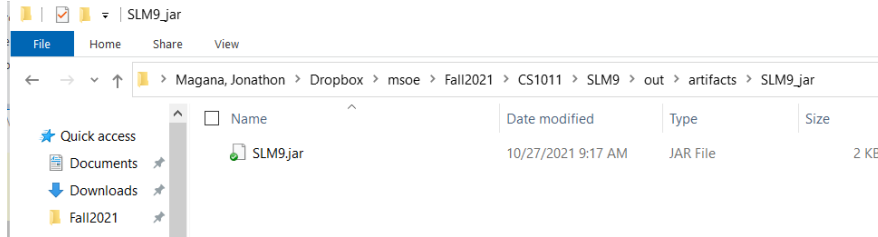
Step 2: From that folder, you can open a Windows command line terminal from the File menu, choosing Windows PowerShell:
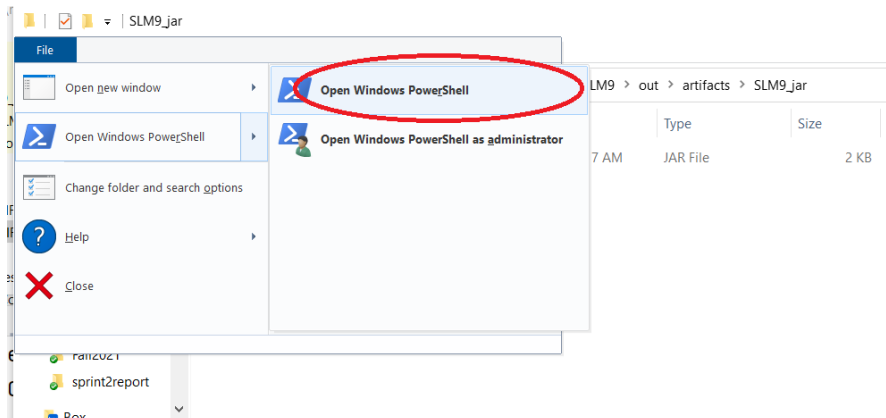
Step 3: You can run various commands here. I run "ls" "dir" and "cd"

You can find more information about commands here: https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/
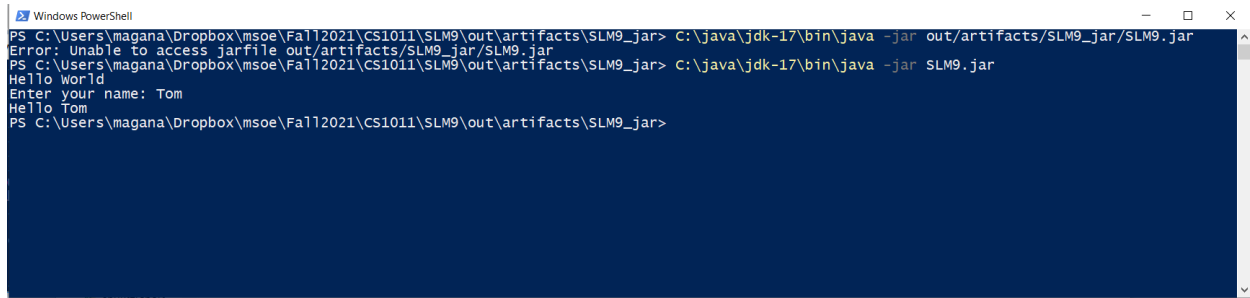
Step 4: We can run our JAR from here by pasting in the same command we used in IntelliJ (STEP 10 above):

C:\java\jdk-17\binjava -jar out/artifacts/SLM9_jar/SLM9.jar

But note that we are already in the …\SLM9\out\artifacts\ directory, so it doesn't work at first. We need to edit the command to account for where we currently are:

C:\java\jdk-17\binjava -jar SLM9.jar

You can also compile your java programs from the command line, using the javac command, see this page for more details: https://docs.oracle.com/javase/7/docs/technotes/tools/windows/javac.html

These instructions and screenshots go over some of the same material that is in these IntelliJ help articles

https://www.jetbrains.com/help/idea/create-jar-from-modules-dialog.html

https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html#package