

# **Working with IntelliJ – Part 3**

**Student Workbook**

Version 2.0



# Table of Contents

<b>Module 1 Working with JAR files in IntelliJ .....</b>	<b>1-1</b>
Section 1-1 Working with JAR Files.....	1-2
Exporting a Jar File in IntelliJ .....	1-3
Creating a Jar File in IntelliJ .....	1-4
Running a Jar File on the Command Line .....	1-7
Viewing the Contents of the Jar File.....	1-8
Exercises .....	1-9



# **Module 1**

## **Working with JAR files in IntelliJ**

## Section 1–1

# Working with JAR Files

# Exporting a Jar File in IntelliJ

---

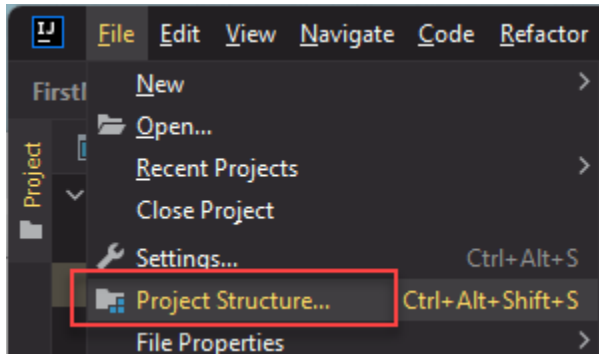
- A Jar file is short for Java Archive. It is merely an enhanced zip or tar file that contains Java class files
- Jar files are portable, and can be transferred over the wire and even be included in other projects
  - They are portable largely because they are a single file, instead of a collection of dozens or more files
- A special type of jar file is a *Runnable Jar File*. What makes it runnable is a **MANIFEST.MF** that is included in the jar file. The **MANIFEST.MF** file contains an entry called **Main-Class** which will tell the Java Runtime Environment what class's main method to start.

**Main-Class:** `com.mycompany.MainApp`

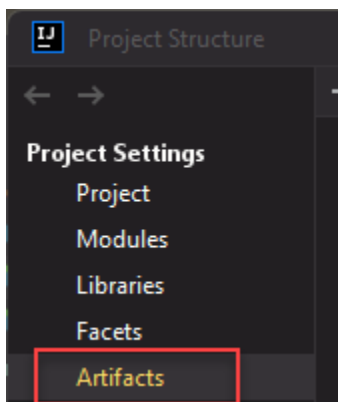
# Creating a Jar File in IntelliJ

---

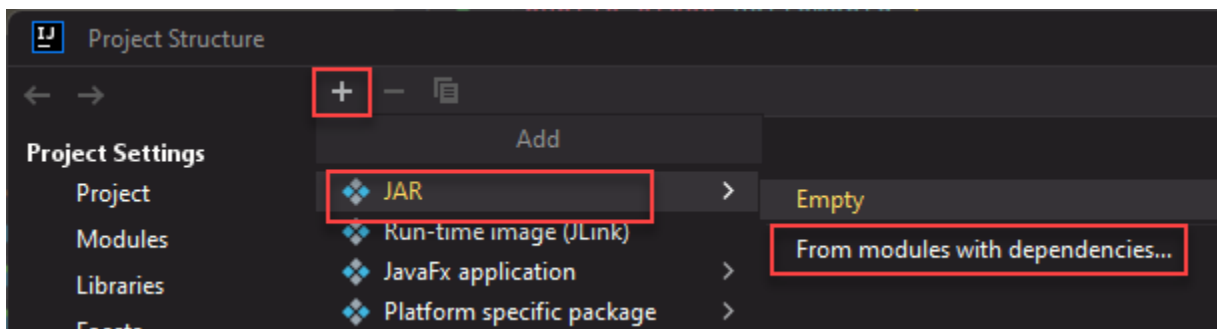
- Click File -> Project Structure



- Select Project Settings -> Artifacts

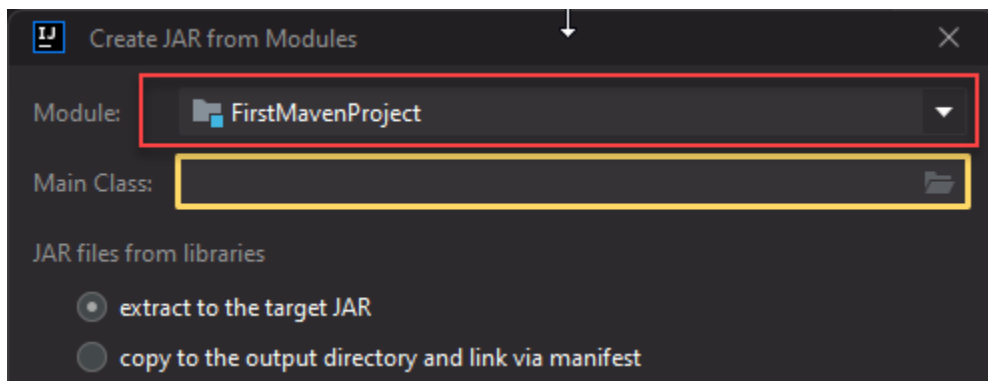


- Click the + to create a new artifact and select Jar -> From modules with dependencies

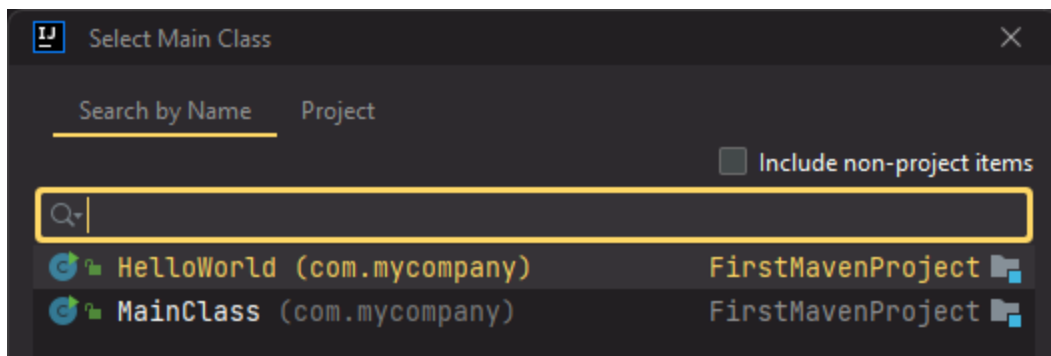




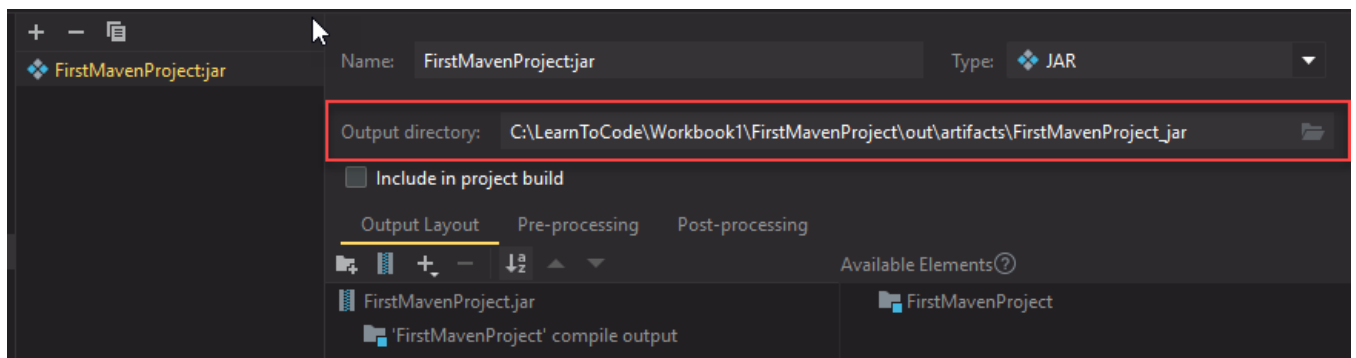
- Ensure that your Project is selected



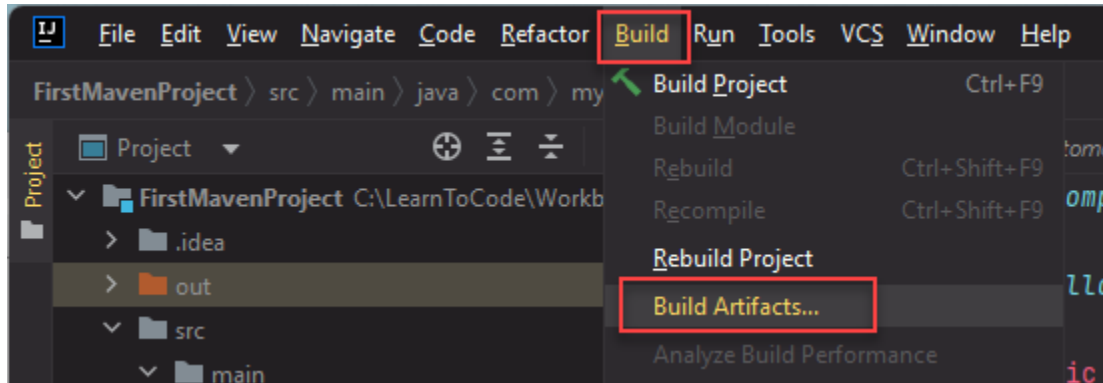
- Select the Main Class (this is the **Entry Point** for the Jar)



- Click **OK**
- This does not generate the Jar, but configures where the Jar will be created



- Click OK to save the configuration
- To generate the Jar go to Build -> Build Artifacts



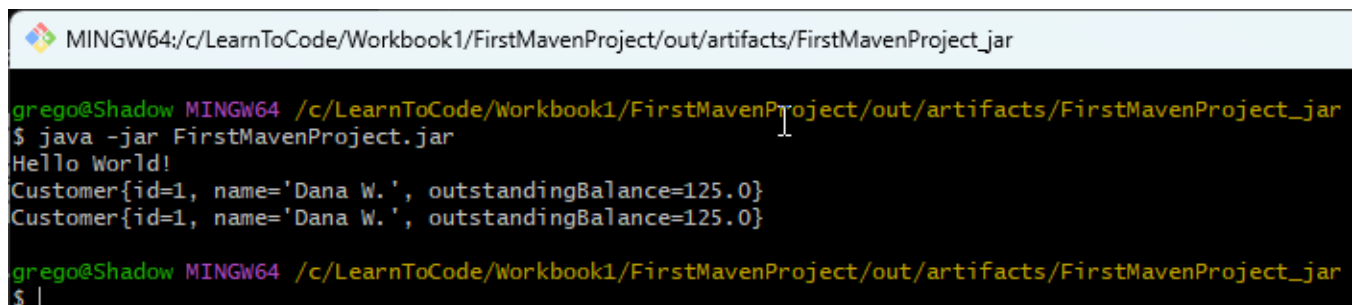
# Running a Jar File on the Command Line

---

- Once you have the Jar file, open a Command Line and run the command

`java -jar <jar-file>`

and you should see the same results as you did in IntelliJ.



```
MINGW64:/c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject_jar
grego@Shadow MINGW64 /c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject_jar
$ java -jar FirstMavenProject.jar
Hello World!
Customer{id=1, name='Dana W.', outstandingBalance=125.0}
Customer{id=1, name='Dana W.', outstandingBalance=125.0}
grego@Shadow MINGW64 /c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject_jar
$ |
```


# Viewing the Contents of the Jar File

---

- To view the contents of the jar file, run the command

```
jar -tvf <jar-file>
```

- Here you will see the manifest file that contains the Main-class declaration



The screenshot shows a terminal window titled "MINGW64:/c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject.jar". The user has entered the command `jar -tvf FirstMavenProject.jar`. The output lists the contents of the jar file, including a manifest file and three class files.

```
grego@Shadow MINGW64 /c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject.jar
$ jar -tvf FirstMavenProject.jar
 63 Mon Mar 13 22:25:06 MDT 2023 META-INF/MANIFEST.MF
  0 Mon Mar 13 22:25:06 MDT 2023 com/
  0 Mon Mar 13 22:25:06 MDT 2023 com/mycompany/
1627 Mon Mar 13 21:23:56 MDT 2023 com/mycompany/Customer.class
 806 Mon Mar 13 21:23:56 MDT 2023 com/mycompany/HelloWorld.class
 571 Mon Mar 13 21:23:56 MDT 2023 com/mycompany/MainClass.class
  0 Mon Mar 13 22:25:06 MDT 2023 META-INF/

grego@Shadow MINGW64 /c/LearnToCode/Workbook1/FirstMavenProject/out/artifacts/FirstMavenProject.jar
$ |
```

- NOTE: You can also extract the .jar file with any Zip software and explore the contents directly

# Exercises

---

## EXERCISE 1

Open the TurtleLogo project that you created in the `workbook-6` directory. Create an executable JAR file. To do this follow the workbook instructions to create a MANIFEST.MF in IntelliJ. This is the file that specifies your application's entry point.

Build and generate the JAR file.

Copy the JAR file to a new folder and run the application. You should be able to copy this to any folder on your computer and run the program from the JAR.