Session : GRADLE

EXERCISE

By: Vishodu Shaozae (EmployeeID: 4174)

1. **Add a gradle dependency and its related repository url.**

   Ans. *Dependency added by including the dependency URL in the dependencies closure and repository in the repositories closure*

```
dependencies {
    compile group: 'com.google.code.gson', name: 'gson', version: '2.8.6'
}
repositories {
   mavenCentral()
}
```

2. **Using java plugin, make changes in the manifest to make the jar executable. Using java -jar JAR_NAME, the output should be printed as "Hello World"**

   Ans. *Made the generated jar executable by providing the path to the entry-point class.*

```
jar{
    manifest {
       attributes(
              "Main-Class": 'App'
       )
    }
}
```

Running *java -jar build/libs/GradleAssignment-1.0-SNAPSHOT.jar*

```
vishodu@Vishodu:~/Documents/GradleAssignment$ java -jar build/libs/GradleAssignment-1.0-SNAPSHOT.jar
Hello World!
```

3. **Differentiate between the different dependency scopes: compile, runtime, testCompile, testRuntime using different dependencies being defined in your build.gradle.**

   Ans. The dependency scopes allow us to limit/define the scope of dependencies in our application. Following are the scopes available:

   a. **testCompile:** Like the name suggests, testCompile specifies the list of dependencies that are needed to compile the test source of our application.
   b. **Compile:** compile on the other hand specifies dependencies needed to build our application
   c. **Runtime:** Runtime provides dependencies that are needed to run our application in the production environment. By default, all dependencies in the compile are automatically available in the runtime scope but not vice versa
   d. **testRuntime:** Similar to testCompile, testRuntime specifies group of dependencies required to perform tests in the runtime environment.

From the below snapshot of the project; junit dependency is used to compile the test sources of the project whereas gson dependency is used during the building of the project

```
dependencies {
    testCompile group: 'junit', name: 'junit', version: '4.12'
    compile group: 'com.google.code.gson', name: 'gson', version: '2.8.6'
}
```

4. **Create a custom plugin which contains a custom task which prints the current date-time. Using that plugin in your project, execute that task after the jar task executes.**

   Ans. Created a plugin module with the showDate task. Overridden the dependOn method to depend on the ‘jar’ task. Setup a buildscript closure referring to the show-date plugin’s dependency thereby connecting the plugin to the main application.

```
buildscript {
   dependencies {
       classpath files('show-date/build/libs/show-date-1.0-SNAPSHOT.jar')
   }
}

apply plugin: 'show-date-plugin'
```

```
vishodu@Vishodu:~/Documents/GradleAssignment$ gradle showDate

> Task :showDate
Current Date and Time: Thu Feb 27 23:04:44 IST 2020
```

5. **Instead of using default source set, use src/main/javaCode1, src/main/javaCode2 to be taken as code source. Make sure that the JAR created contains files from both the directories and not from src/main/java.**

   Ans. Created the two source directories and added them to the sourceSets.

```
sourceSets{
    main {
        java {
            srcDirs = ['src/main/javaCode1', 'src/main/javaCode2']
        }
    }
}
```

```
vishodu@Vishodu:~/Documents/GradleAssignment$ ls src/main/
javaCode1  javaCode2
vishodu@Vishodu:~/Documents/GradleAssignment$ jar tvf build/libs/GradleAssignment-1.0-SNAPSHOT.jar
     0 Thu Feb 27 23:03:08 IST 2020 META-INF/
    42 Thu Feb 27 23:03:08 IST 2020 META-INF/MANIFEST.MF
   270 Thu Feb 27 23:03:08 IST 2020 InsideJavaCode1.class
   533 Thu Feb 27 23:03:08 IST 2020 Process.class
   270 Thu Feb 27 23:03:08 IST 2020 InsideJavaCode2.class
   513 Thu Feb 27 23:03:08 IST 2020 App.class
```

6. **Override the Gradle Wrapper task to install a different version of gradle. Make sure that the task written in Q4 also executes with it.**

   Ans. Below snapshots show

   a. Current gradle version

b. Over-riding gradle wrapper class to install gradle version 4.4.1

c. Running the showDate task with gradlew

```
vishodu@Vishodu:~/Documents/GradleAssignment$ gradle wrapper --version

------------------------------------------------------------
Gradle 6.2.1
------------------------------------------------------------

Build time:   2020-02-24 20:24:10 UTC
Revision:     aacbcb7e587faa6a8e7851751a76183b6187b164

Kotlin:       1.3.61
Groovy:       2.5.8
Ant:          Apache Ant(TM) version 1.10.7 compiled on September 1 2019
JVM:          1.8.0_242 (Azul Systems, Inc. 25.242-b20)
OS:           Linux 4.15.0-76-generic amd64
```

```
vishodu@Vishodu:~/Documents/GradleAssignment$ ./gradlew showDate
Downloading https://services.gradle.org/distributions/gradle-6.2.1-bin.zip
.........10%.........20%.........30%.........40%.........50%.........60%.........70%.........80%.........90%.........100%

> Task :showDate
Current Date and Time: Thu Feb 27 23:14:14 IST 2020

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.2.1/userguide/command_line_interface.html#sec:command_line_warnings
```

7. **Run the gradle profile command and attach the resulting files.**

   Ans. Ran the '*gradle build --profile'* command. Attached the report in the build directory.