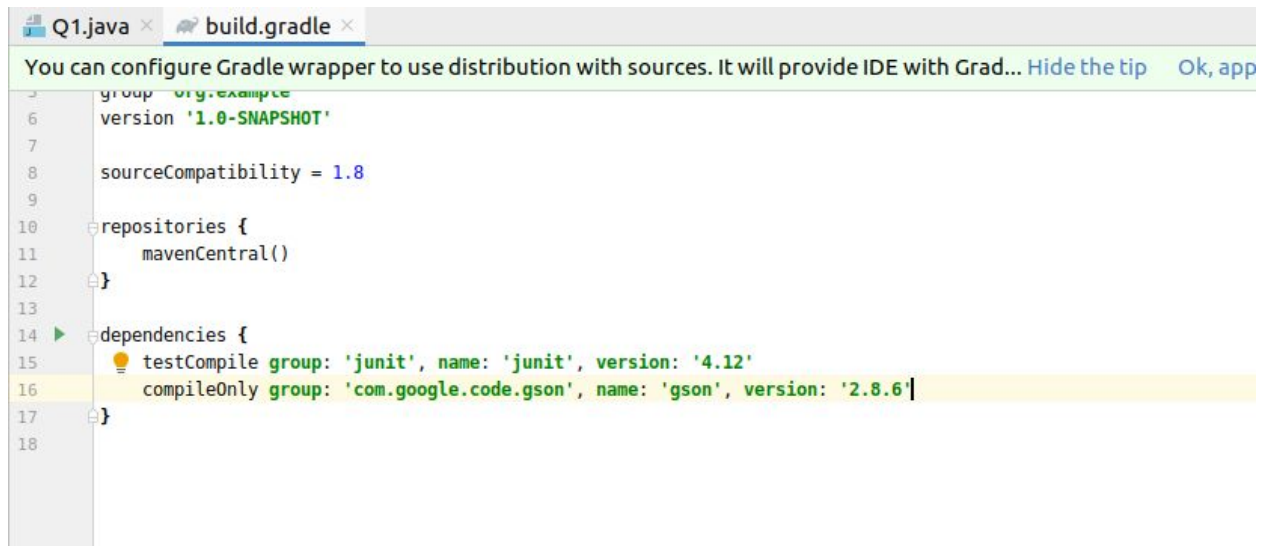


Q1 Add a gradle dependency and its related repository url.

Ans <https://mvnrepository.com/artifact/com.google.code.gson/gson>

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



Q2 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"

apply plugin : 'java'

```
sourceSets{  
    main{  
        java{  
            srcDirs = ["src/main/java"]  
        }  
    }  
}
```

```
task sd {  
    dependsOn(build)  
    doLast {  
        println new Date();  
    }  
}
```

```
jar{
```

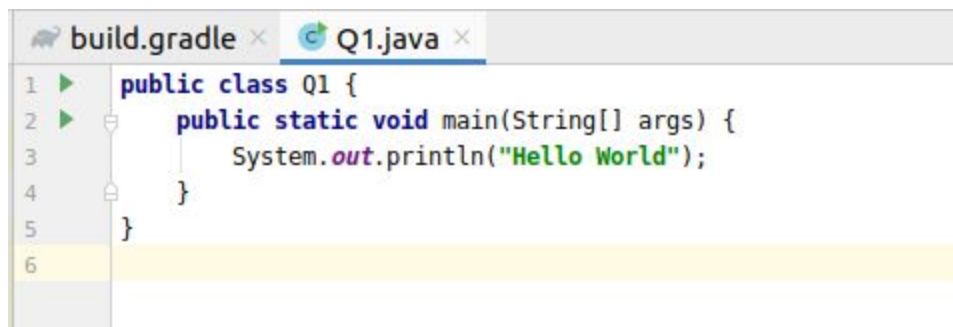
```

manifest{
    attributes(
        "Main-Class": 'Q1',
        "Class-Path": configurations.compile.collect { it.getName() }.join(' '),
    )
}

repositories {
    mavenCentral()
}

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

```



```

build.gradle x Q1.java x
1 public class Q1 {
2     public static void main(String[] args) {
3         System.out.println("Hello World");
4     }
5 }
6

```

Output

```

tushar@tushar:~/gradle$ java -jar /home/tushar/gradle/build/libs/gradle1.jar
Hello World
tushar@tushar:~/gradle$ 

```

Q3 Differentiate between the different dependency scopes: compile, runtime, testCompile, testRuntime using different dependencies being defined in your build.gradle.

Compile: The compile scope is the default scope. We can use it when we have no special requirements for declaring a certain dependency.

Runtime: We use the runtime scope for dependencies that are not needed at compile time, like when we're compiling against an API and only need the implementation of that API at runtime.

testCompile: We can use the testCompile scope for dependencies that are only needed in tests compile time and that should not be available in production code.

testRuntime: We use the runRuntime scope for dependencies that are not needed at compile time, like when we're compiling against an API and only need the implementation of that API at runtime.

Q4 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.

The screenshot displays an IDE interface with a project explorer on the left, a central editor showing a `build.gradle` file, and a bottom panel showing the execution output of a Gradle task.

Project Explorer: The project structure includes a `gradle` directory with sub-directories `.gradle`, `.idea`, `build` (containing `classes`, `generated`, and `libs`), and `tmp`. The `gradle` directory also contains `gradle1.jar`. A `gradle` wrapper is present, including `gradle-wrapper.jar` and `gradle-wrapper.properties`. The `src` directory contains a `main` sub-directory.

build.gradle: The file contains the following configuration:

```
1  apply plugin : 'java'
2
3  sourceSets {
4      main {
5          java {
6              srcDirs = ["src/main/java"]
7          }
8      }
9  }
10
11 task sd {
12     dependsOn(build)
13     doLast {
14         println new Date();
15     }
16 }
17
18 jar {
```

Run: gradle [sd]

The execution output shows the following sequence of tasks and results:

```
> Task :check UP-TO-DATE
> Task :build

> Task :sd
Mon Mar 02 15:44:43 IST 2020

BUILD SUCCESSFUL in 0s
3 actionable tasks: 3 executed
3:44:43 PM: Task execution finished 'sd'.
```

Q5 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`.

```
build.gradle x Q5.java x Q1.java x
1  import lombok.Getter;
2  import lombok.Setter;
3  import lombok.ToString;
4
5  @Getter
6  @Setter
7  @ToString
8
9  public class Q5{
10      String name ;
11      int age ;
12      int id ;
13
14  }
15
```

```
build.gradle x Q5.java x Q1.java x
1  public class Q1 {
2      public static void main(String[] args) {
3          Q5 obj1 = new Q5();
4          obj1.name = "tushar";
5          obj1.age = 33;
6          obj1.id = 4044;
7          System.out.println(obj1);
8      }
9  }
10
```

```
BUILD SUCCESSFUL in 0s
2 actionable tasks: 2 executed
tushar@tushar:~/gradle$ java -jar /home/tushar/gradle/build/libs/gradle1.jar
Q5(name=tushar, age=33, id=4044)
tushar@tushar:~/gradle$
```

Terminal Build 4: Run 6: TODO

Q6 Override the Gradle Wrapper task to install a different version of gradle. Make sure that the task written in Q4 also executes with it.

```
tushar@tushar:~/gradle$ ./gradlew --version
Downloading https://services.gradle.org/distributions/gradle-4.4.1-bin.zip
.....
Unzipping /home/tushar/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8ilv3zqx4q949t2x/gradle-4.4.1-bin/46gopw3g8ilv3zqx4q949t2x
Set executable permissions for: /home/tushar/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8ilv3zqx4q949t2x/gradle-4.4.1/bin/gradle

-----
Gradle 4.4.1
-----

Build time: 2017-12-20 15:45:23 UTC
Revision: 10ed9dc355dc39f6307cc98fbd8cea314bdd381c

Groovy: 2.4.12
Ant: Apache Ant(TM) version 1.9.9 compiled on February 2 2017
JVM: 1.8.0_242 (Azul Systems, Inc. 25.242-b20)
```

```
tushar@tushar:~/gradle$ gradle wrapper --gradle-version 6.2.1
:wrapper

BUILD SUCCESSFUL in 0s
1 actionable task: 1 executed
tushar@tushar:~/gradle$ ./gradlew --version

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

Build time: 2020-02-24 20:24:10 UTC
Revision: aacbc7e587faa6a8e7851751a76183b6187b164

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
```

```
tushar@tushar:~/gradle$ java -jar /home/tushar/gradle/build/libs/gradle1.jar
hello world
tushar@tushar:~/gradle$
```

Terminal Build 4: Run 6: TODO

Q7 Run the gradle profile command and attach the resulting files.

Profile report

Profiled build: help

Started on: 2020/03/02 - 16:26:30

Summary

Configuration

Dependency Resolution

Task Execution

Description	Duration
Total Build Time	0.371s
Startup	0.349s
Settings and BuildSrc	0.001s
Loading Projects	0.001s
Configuring Projects	0.011s
Task Execution	0.001s