1) Look up java plugin documentation. Make changes in manifest to make it executable with correct class. When run using java -jar JAR_NAME_HERE the output should be text "Hello World" on the console.

java plugin documentation:
https://docs.gradle.org/current/userguide/java_plugin.html

build.gradle

```
apply plugin: "java"
sourceSets {
  main {
      java {
          srcDirs = ['src/main/java']
      }
  }
}


version='1.0'
jar {
  manifest {
      attributes 'Implementation-Title': 'Gradle Tutorial',
              'Implementation-Version': version,
              'Main-Class': 'Application'
  }
}

task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
  main = "Application"
  classpath = sourceSets.main.runtimeClasspath
}
```

Application.java

```java
public class Application {
  public static void main(String[] args) {
      System.out.println("Hello World");
  }
}
```

```
Terminal
File  Edit  View  Search  Terminal  Help
ttn@ttn:~ $ mkdir Assignment_Gradle
ttn@ttn:~ $ cd Assignment_Gradle
ttn@ttn:Assignment_Gradle $ gradle -v

------------------------------------------------------------
Gradle 4.10.2
------------------------------------------------------------

Build time:   2018-09-19 18:10:15 UTC
Revision:     b4d8d5d170bb4ba516e88d7fe5647e2323d791dd

Kotlin DSL:   1.0-rc-6
Kotlin:       1.2.61
Groovy:       2.4.15
Ant:          Apache Ant(TM) version 1.9.11 compiled on March 23 2018
JVM:          1.8.0_202 (Amazon.com Inc. 25.202-b08)
OS:           Linux 4.15.0-45-generic amd64

ttn@ttn:Assignment_Gradle $ gradle init

BUILD SUCCESSFUL in 0s
2 actionable tasks: 2 executed
ttn@ttn:Assignment_Gradle $ gradle build

BUILD SUCCESSFUL in 0s
```

```
ttn@ttn:Assignment_Gradle $ java -jar build/libs/Assignment_Gradle-1.0.jar
Hello World
```

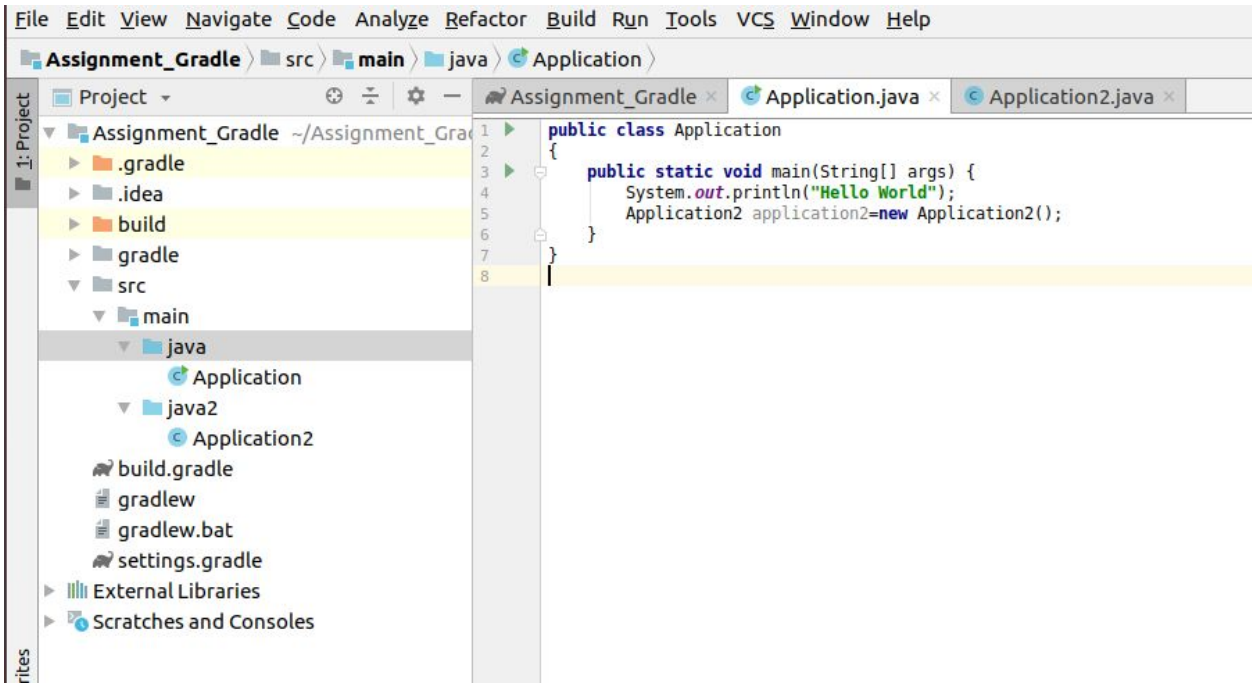
2) look up idea plugin. make changes in build.gradle so that the sources of src/main/java as well as src/main/java2 are taken as sources. Ensure that when you make JAR file class files in both are added to the JAR. This will teach you how projects with non-conventional structure can be used with gradle.

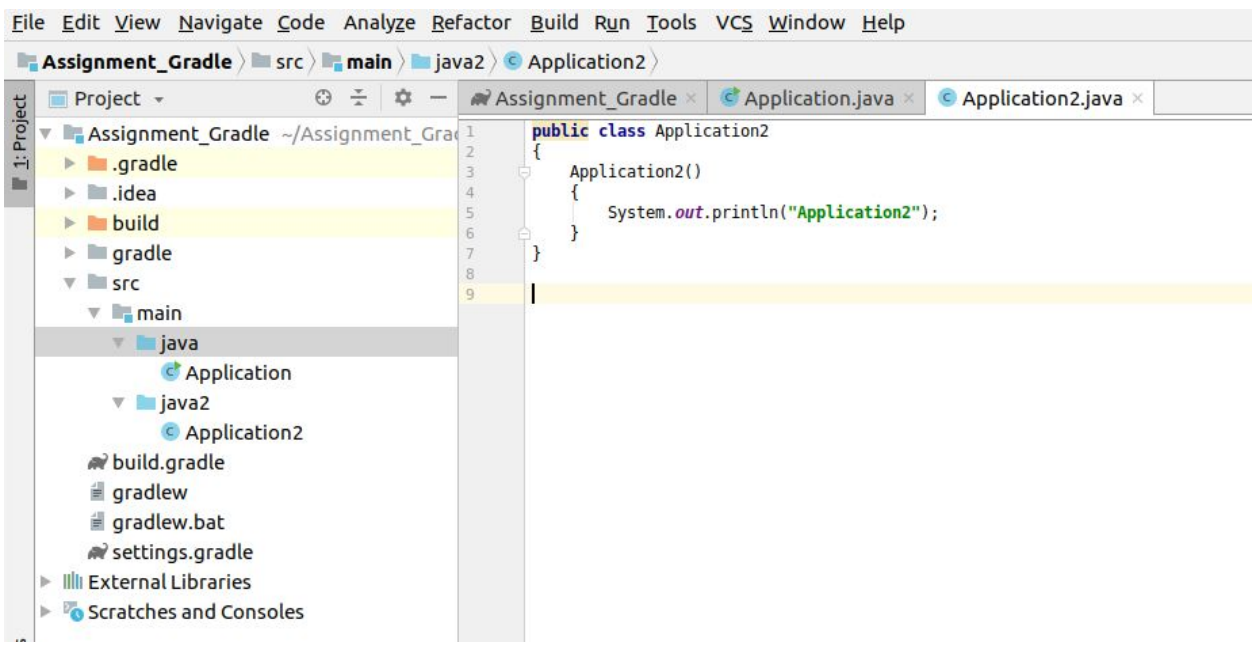
IDEA plugin documentation;
https://docs.gradle.org/current/userguide/idea_plugin.html

build.gradle

```
apply plugin: "java"
sourceSets {
  main {
    java {
      srcDirs = ['src/main/java', 'src/main/java2']
    }
  }
}


version='1.0'
jar {
  manifest {
    attributes 'Implementation-Title': 'Gradle Tutorial',
          'Implementation-Version': version,
          'Main-Class': 'Application'
  }
}

task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
  main = "Application"
  classpath = sourceSets.main.runtimeClasspath
```

```
}
```





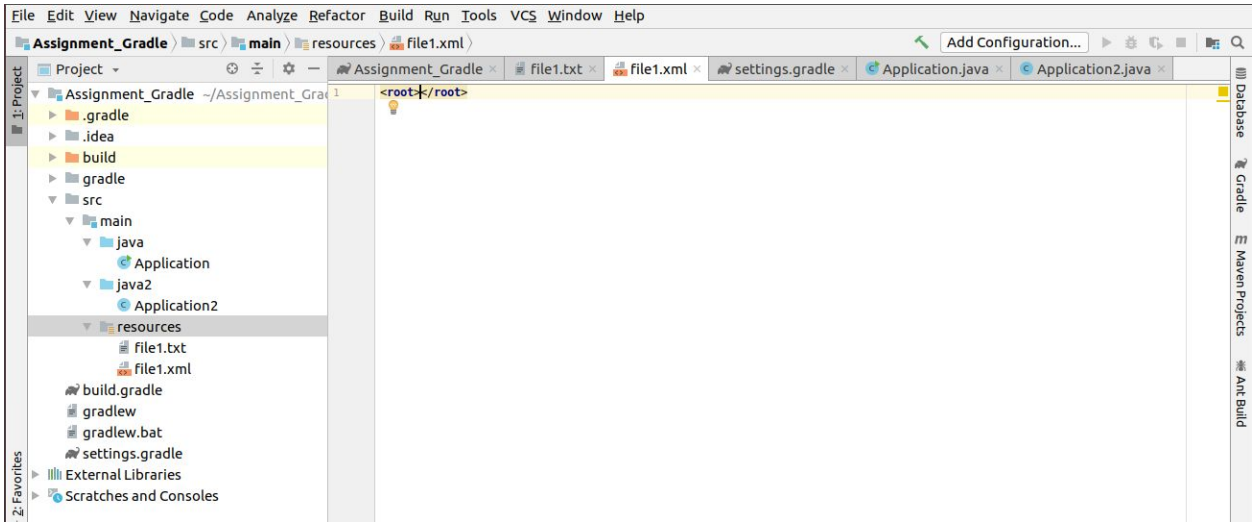

```
ttn@ttn:Assignment_Gradle $ gradle build

BUILD SUCCESSFUL in 0s
2 actionable tasks: 2 executed
```

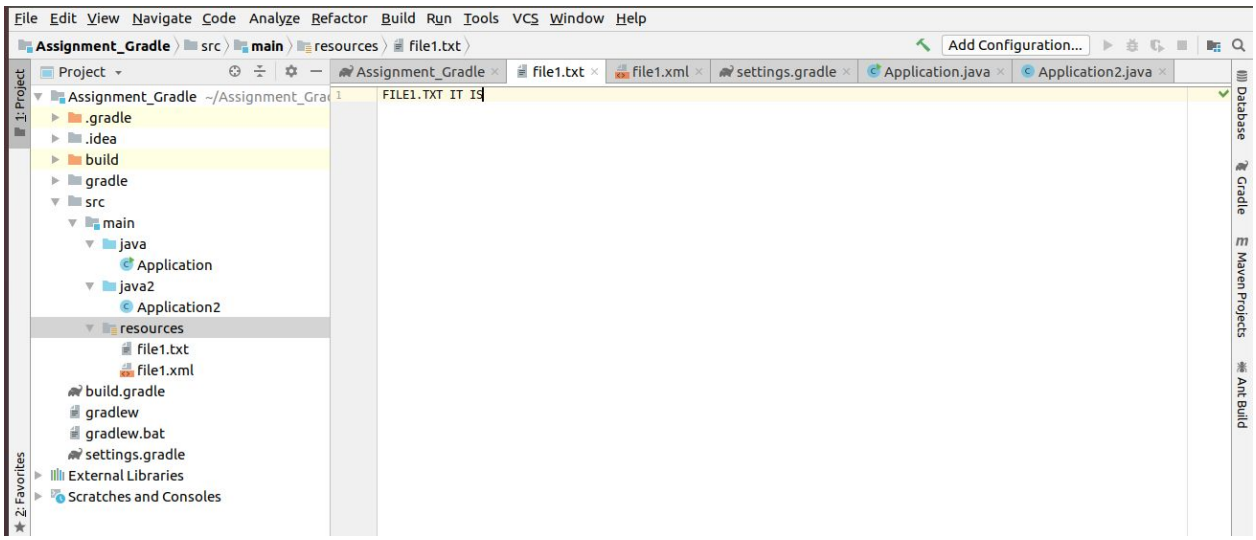
```
ttn@ttn:Assignment_Gradle $ java -jar build/libs/Assignment_Gradle-1.0.jar
Hello World
Application2
```

3) add 2 files file1.xml and file1.txt in src/main/resources manually. make changes so that when creating jar only file1.xml is added to the jar.

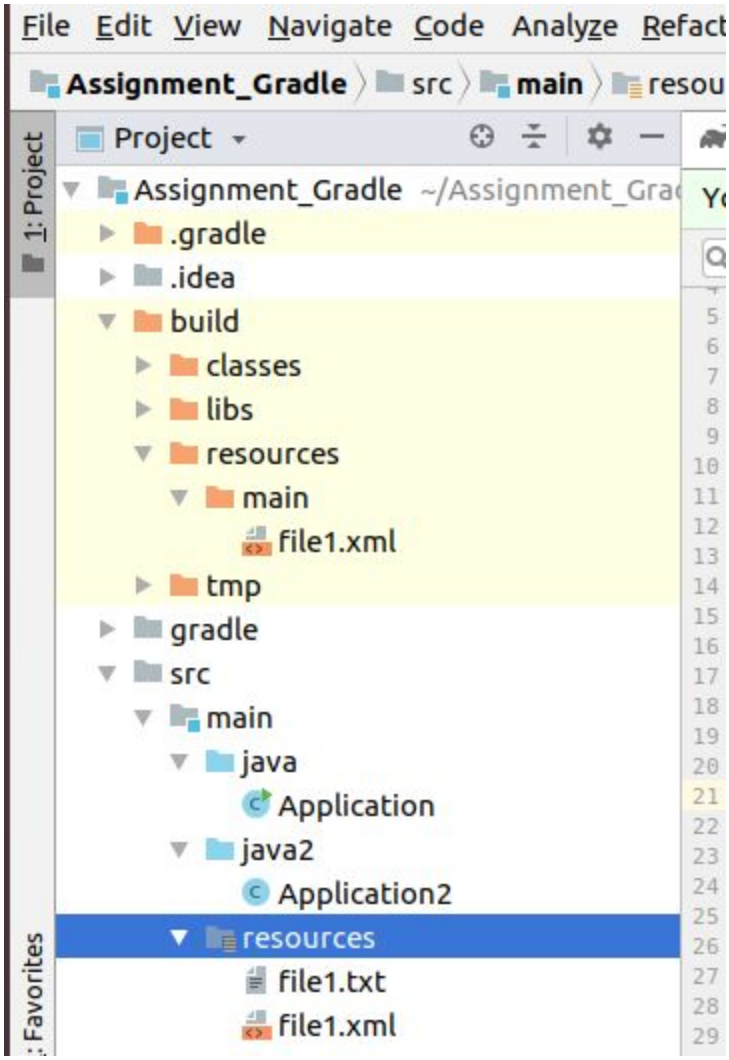
build.gradle

```
apply plugin: "java"
sourceSets {
  main {
     java {
        srcDirs = ['src/main/java', 'src/main/java2']
     }
     resources {
        srcDirs = ['src/main/resources']
        exclude 'file1.txt'
     }
  }
}


version='1.0'
jar {
  manifest {
     attributes 'Implementation-Title': 'Gradle Tutorial',
           'Implementation-Version': version,
           'Main-Class': 'Application'
  }
}

task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
  main = "Application"
  classpath = sourceSets.main.runtimeClasspath
}
```






```
ttn@ttn:Assignment_Gradle $ java -jar build/libs/Assignment_Gradle-1.0.jar
Hello World
Application2
ttn@ttn:Assignment_Gradle $
```


File Edit View Navigate Code Analyze Refact
Assignment_Gradle src main resou
1: Project
Project
Assignment_Gradle ~/Assignment_Grad
.gradle
.idea
build
classes
libs
resources
main
file1.xml
tmp
gradle
src
main
java
Application
java2
Application2
resources
file1.txt
file1.xml
: Favorites

4) find how to what is an uberjar. Make changes so you can use commons lang3 StringUtil in your jar. Make this uber jar executable. The output should be text but that should be using the StringUtils class of commons lang3

Fat/uber jar: https://www.baeldung.com/gradle-fat-jar

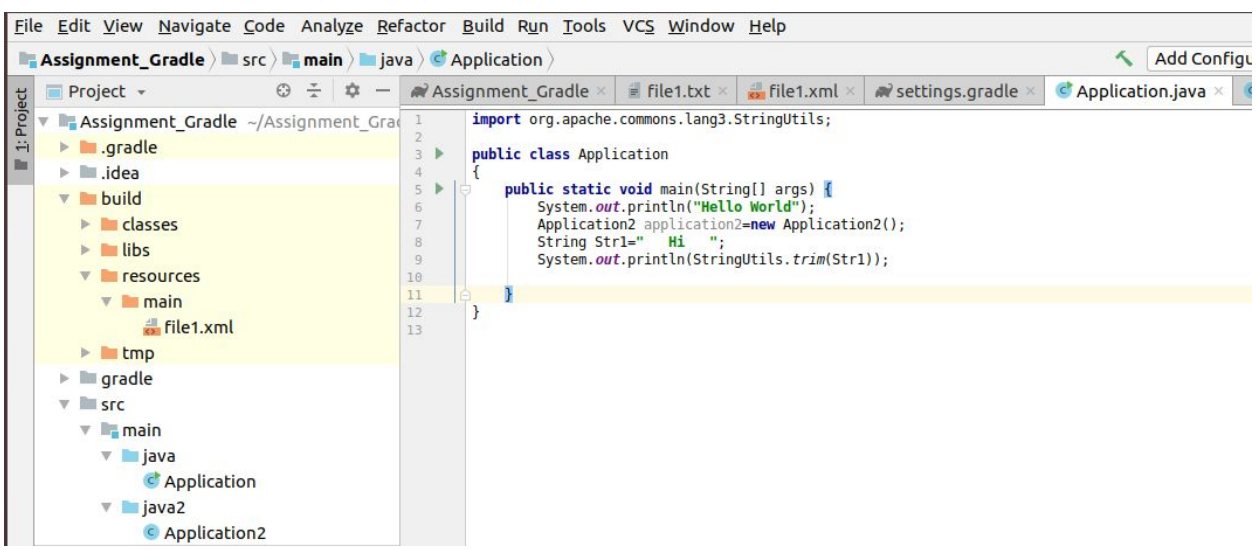
build.gradle

```
apply plugin: "java"
sourceSets {
  main {
    java {
      srcDirs = ['src/main/java', 'src/main/java2']
    }
    resources {
      srcDirs = ['src/main/resources']
      exclude 'file1.txt'
    }
  }
}
```

```
repositories {
  maven { url "https://repo.maven.apache.org/maven2" }
}

dependencies {
  compile group: 'org.apache.commons', name: 'commons-lang3', version: '3.8.1'
}
```

```
version = '1.0'
jar {
  manifest {
     attributes 'Implementation-Title': 'Gradle Tutorial',
           'Implementation-Version': version,
           'Main-Class': 'Application'
  }
  from {
     (configurations.compile).collect { it.isDirectory() ? it : zipTree(it) }
  }
}

task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
  main = "Application"
  classpath = sourceSets.main.runtimeClasspath
}
```

```
ttn@ttn:Assignment_Gradle $ gradle build

BUILD SUCCESSFUL in 0s
3 actionable tasks: 2 executed, 1 up-to-date
```



```
ttn@ttn:Assignment_Gradle $ gradle build

BUILD SUCCESSFUL in 0s
3 actionable tasks: 2 executed, 1 up-to-date
ttn@ttn:Assignment_Gradle $ gradle -q runJavaApplication
Hello World
Application2
Hi
ttn@ttn:Assignment_Gradle $
```

5) Find a maven repository and add it as a repository. You can use bintray, jcenter

build.gradle

```
apply plugin: "java"
apply from: "mytasks.gradle"
sourceSets {
  main {
    java {
      srcDirs = ['src/main/java', 'src/main/java2']
    }

    resources {
      srcDirs = ['src/main/resources']
      exclude 'file1.txt'
    }


  }
}




repositories {
  mavenCentral()
  jcenter()
  maven { url "https://repo.maven.apache.org/maven2" }
}


dependencies {
  compile group: 'org.apache.commons', name: 'commons-lang3', version: '3.8.1'
}
```

```
version = '1.0'
jar {
  manifest {
    attributes 'Implementation-Title': 'Gradle Tutorial',
               'Implementation-Version': version,
               'Main-Class': 'Application'
  }





  from {
    (configurations.compile).collect { it.isDirectory() ? it : zipTree(it) }
  }
}




  task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
    main = "Application"
    classpath = sourceSets.main.runtimeClasspath
  }
```

```
BUILD SUCCESSFUL in 0s
3 actionable tasks: 3 up-to-date
ttn@ttn:Assignment_Gradle $
```

6) Write a task in file "mytasks.gradle" and use it in your **build.gradle**

```
apply plugin: "java"
apply from : "mytasks.gradle"
sourceSets {
  main {
     java {
        srcDirs = ['src/main/java', 'src/main/java2']
     }
     resources {
        srcDirs = ['src/main/resources']
        exclude 'file1.txt'
     }
  }
}




repositories {

  mavenCentral()
  jcenter()
  maven { url "https://repo.maven.apache.org/maven2" }
}

dependencies {
  compile group: 'org.apache.commons', name: 'commons-lang3', version: '3.8.1'
}



version = '1.0'
jar {
  manifest {
     attributes 'Implementation-Title': 'Gradle Tutorial',
           'Implementation-Version': version,
```

```
            'Main-Class': 'Application'
    }
    from {
        (configurations.compile).collect { it.isDirectory() ? it : zipTree(it) }
    }
}

task runJavaApplication(type: JavaExec, dependsOn: 'classes') {
    main = "Application"
    classpath = sourceSets.main.runtimeClasspath
}
```

mytasks.gradle

```
task myTask <<{
    println "This is MyTask"
}
```

```
ttn@ttn:Assignment_Gradle $ gradle build

Deprecated Gradle features were used in this build, making it incompatible with Gradle 5.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/4.10.2/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 0s
3 actionable tasks: 3 up-to-date
```

```
ttn@ttn:Assignment_Gradle $ gradle -q myTask
This is MyTask
ttn@ttn:Assignment_Gradle $
```