**COP2806C Module 1 Programming Assignment Part 2 - Introduction to Spring Boot**

In this assignment we will implement the Spring Boot version of the Hello World web service application as described in Ch. 1 of our Spring textbook. Follow the instructions below to create the project and execute the application, then submit your project folder to the GitHub classroom repo (include the two snips described in the last step). There is no Canvas submission for this assignment.

We will continue to work from the command line.

I would recommend that you use the Horizon system for this assignment as all software is installed and preconfigured. If you use Horizon, download this instruction document on that system to insure you can copy and paste as necessary.

You will need Gradle, Java 17, and an Internet connection for this assignment.

You can clone your repo from Programming Assignment Part 1and copy the contents to your new GitHub classroom repo to start with. Be sure that you **do not** copy the **.git** folder from your cloned repo.

1. Move to the isf6 folder that you created for the Part 1 assignment.

```
C:\Users\YourUserID\DownLoads>cd isf6
```

2. Create a new subfolder named "chapter01-hello-boot"

```
C:\Users\YourUserID\DownLoads\isf6>mkdir chapter01-hello-boot
```

3. Move to the new subfolder and create a **build.gradle** file with the following contents:

```
plugins {
    id "application"
    id 'org.springframework.boot' version '3.0.0-M4'
    id 'io.spring.dependency-management' version '1.0.13.RELEASE'
}
dependencies {
    implementation \
    'org.springframework.boot:spring-boot-starter-web'
}
application {
    mainClass.set("chapter01.HelloWorldController")
}
```

4. In the top-level isf6 folder, modify the settings.gradle file by adding the line shown in bold below:

```
pluginManagement {
    repositories {
        maven {
            url "https://repo.spring.io/milestone"
        }
        maven {
```

```
            url "https://repo.spring.io/snapshot"
            mavenContent {
                snapshotsOnly()
            }
        }
        gradlePluginPortal()
    }
}
rootProject.name = 'isf6'
include 'chapter01-hello-world'
include 'chapter01-hello-boot'
```

5. Move to the subfolder and create the folder structure as shown below. Remember that you can create a multi-level folder hierarchy by using a single mkdir command.

```
C:\Users\YourUserID\DownLoads\isf6>cd chapter01-hello-boot
C:\...\isf6\chapter01-hello-boot>mkdir src\main\java\chapter01
```

6. In the **chapter01** folder at the bottom of the folder structure create the following Java class (**HelloWorldController.java**)

```
package chapter01;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@EnableAutoConfiguration
public class HelloWorldController {
  @GetMapping(path = "/", produces = "text/plain")
  @ResponseBody
  public String getMessage() {
    return "Hello, world!";
  }
  public static void main(String[] args) {
    SpringApplication.run(HelloWorldController.class,args);
  }
}
```

7. Run the project using **gradle :chapter01-hello-boot:run** from the `isf6 folder`:

```
C:\...\isf6>gradle :chapter01-hello-boot:run
```

It will take a minute to start the server. When it is running you will see the Spring banner; at that point you can open a browser and connect to **localhost:8080**.

You should see the Hello World message in the browser. Snip the banner from the server window and snip the top left of the browser window showing the Hello World message and include these files as .jpeg (or .jpg)  in the top-level folder of your GitHub submission.