The University of the West Indies, St. Augustine
COMP 3607 Object Oriented Programming II
2020/2021 Semester 1
Lab Tutorial - Week 11

This tutorial focuses on object persistence.

Learning Objectives:

- Write code to read import data from a JSON file and deserialise as a Java Object class
- Write code to serialise a Java Object class as JSON and produce a JSON file
- Write code to read import data from an XML file and deserialise as a Java Object class
- Write code to serialise a Java Object class as XML and produce an XML file

Resources/ Useful Links:

- https://maven.apache.org/guides/mini/guide-naming-conventions.html
- http://tutorials.jenkov.com/java-json/jackson-installation.html

Instructions

Download the Java classes for Week 11 (retrieve from myElearning). We will set up Maven projects for the two parts of the lab.

Activities - Part A: Serialization

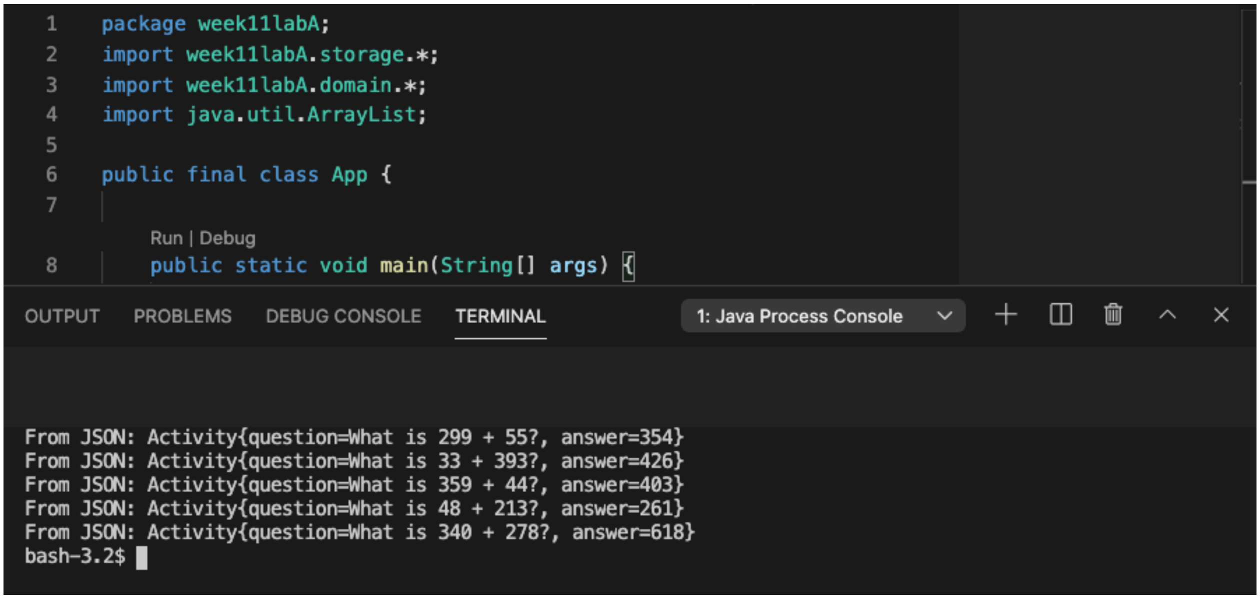
1. Download the Maven project for your work from myElearning and set it up using the VS Code editor.

2. Add the jackson dependencies to the pom.xml files (http://tutorials.jenkov.com/java-json/jackson-installation.html)

3. Create a new class, `JSONActivityExporter`, that:
   - Implements the `ActivityExporter` interface in the Storage folder
   - Exports the `Activity` class as a JSON file
   - Stores the JSON file in the `files` folder
   - Uses the `getFilename( )` method of the `Activity` class to generate the first part of filenames. E.g. the first Activity object should be serialised and stored in `activity0.json`

4. Evaluate the functionality of your code from step 3 by creating 5 activity objects and verifying that the objects are properly serialised. You should see 5 JSON files created in the files folder (see Figure 3)
5. Create a new class, `JSONActivityImporter`, that:
   - Implements the `ActivityImporter` interface in the Storage folder
   - Imports a JSON file and deserialises the data into an `Activity` class object

6. Evaluate the functionality of your code from step 3 by printing the details of the deserialised objects. Verify that the contents of the files match the output.

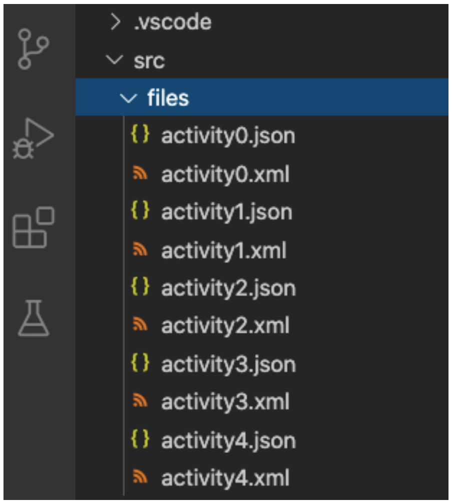


```java
package week11labA;
import week11labA.storage.*;
import week11labA.domain.*;
import java.util.ArrayList;

public final class App {

    Run | Debug
    public static void main(String[] args) {
```

```
From JSON: Activity{question=What is 299 + 55?, answer=354}
From JSON: Activity{question=What is 33 + 393?, answer=426}
From JSON: Activity{question=What is 359 + 44?, answer=403}
From JSON: Activity{question=What is 48 + 213?, answer=261}
From JSON: Activity{question=What is 340 + 278?, answer=618}
bash-3.2$
```

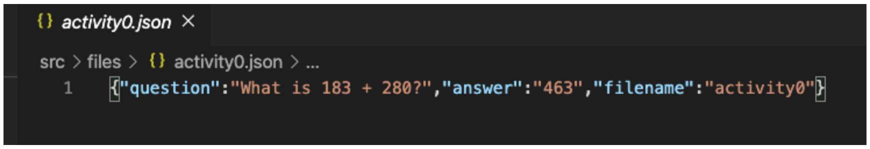
Figure 1. Sample JSON content

7. Repeat steps 3-4 for an XMLActivityExporter class.
8. Repeat steps 5-6 for an XMLActivityImporter class.
9. Examine what happens when you remove the XML annotations from the Activity class.
10. Examine what happens when you remove an accessor/mutator from the Activity class.



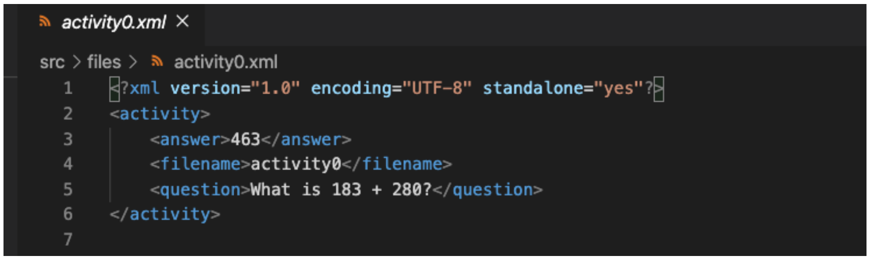
```
> .vscode
∨ src
  ∨ files
    activity0.json
    activity0.xml
    activity1.json
    activity1.xml
    activity2.json
    activity2.xml
    activity3.json
    activity3.xml
    activity4.json
    activity4.xml
```

Figure 3. List of final output files



```json
{"question":"What is 183 + 280?","answer":"463","filename":"activity0"}
```

Figure 2. JSON file content



```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<activity>
    <answer>463</answer>
    <filename>activity0</filename>
    <question>What is 183 + 280?</question>
</activity>

```

Figure 4. XML file content