Programming Test Re-run 2018

Rules

- READ IT ALL FIRST (you might find it easier to do some questions in parallel)
- You can use the Internet. You need to refer to any code taken from the web (not necessary for code I have given). Not doing so is plagiarism and will be treated severely.
- No communication between students is allowed.
- You can ask questions, I decide if they are fair in an exam setting.

The What

You are required to create a web service, which provides URL end points for CRUD operations on Student resources. This is to be done using Jersey.

The scope of this assessment does NOT include databases or persistence, as such you will NOT actually delete, or save. You will just return appropriate messages as response to requests. The web service will return ONLY JSON. The web service accepts, where appropriate, JSON input. You are required to use GSON explicitly through the code, you will not use the framework to convert the POJOs to JSON.

The Specifics

1. **Create a service to return a list of all the Students (25%)**

   Create an ArrayList storing five Student instances. When the request for a list of users is invoked, return this list to the caller in JSON format. The list returned will be filtered from the full list by an offset i.e. the user supplies a "3" then the returned list is offset from main list. This should be contained in an annotated method. The Student class must be defined by yourself; a student should have a name, an age, and a list of subjects which they are taking. If no offset is provided the full list should be returned.

2. **Create a service to search for Students (20%)**

   Use the same list as before, if the user(s) specified by their name (this is a query) are present, return those users in JSON format, otherwise return a sensible HTTP error code and JSON output. This should be contained in an annotated method. The user should be retrieved from the ArrayList. The user should be able to search by first name, surname, and full name.

3. **Create a service to update a Student (15%)** On the server there should be a method which updates a Student object from the query parameter input. if the student was present in the list, then the new object will replace the existing one, if the student did not exist then the update should fail. Return a sensible response, with the URL of the resource.

4. **Create a service to delete a Student (20%)** Examine the input parameter and return a message indicating that the resource was removed. You will return an affirmative message stating that the resource is deleted only if the user is in our ArrayList. This should all be contained in a properly annotated method.

5. **cURL Client (20%)**

   Provide cURL commands to invoke each of the above defined HTTP requests. These should be included as comments above each method. I will use these to test your code.