Lab Assignment: Parsing JSON Response Body with Rest Assured

Objective:
Learn how to parse JSON response bodies using Rest Assured in BDD style. You'll work with both simple and nested JSON responses and implement methods to extract specific data points.

Tasks:

1. **Simple JSON Parsing**
   Parse a simple JSON response body to retrieve the value of `place_id`.

   JSON Response:

```
{
    "status": "OK",
    "place_id": "9172e95034d47483d8e0775710eb6a54",
    "scope": "APP",
    "reference":
"de2fad9729217c7b69d9a0bb4906e742de2fad9729217c7b69d9a0bb4906e742",
    "id": "de2fad9729217c7b69d9a0bb4906e742"
}
```

   Task: Write a test that retrieves the value of `place_id` from the response.

---

2. **Nested JSON Parsing**
   Parse a nested JSON response to retrieve specific values based on the following scenarios.

   JSON Response:

```
{
    "dashboard": {
        "purchaseAmount": 1060,
        "website": "code2test.com"
    },
    "courses": [
        {
            "title": "Selenium",
            "price": 50,
            "copies": 6
        },
        {
            "title": "Appium",
            "price": 40,
            "copies": 4
        },
        {
            "title": "Rest Assured",
            "price": 45,
            "copies": 10
        },
        {
            "title": "Postman",
            "price": 30,
            "copies": 5
        }
    ]
}
```

Scenarios:
a) Get the purchase amount of the course.
b) Get the title of the first course (`Selenium`).
c) Get the total copies sold for `Postman`.
d) Count the total number of courses.
e) Print all the course titles.

---

3. **Bonus Challenge:**
   Write assertions to validate that the extracted values match the expected values.

Expected Output:

1. The value of `place_id` from the simple JSON will be printed and validated.
2. For the nested JSON:
   - Purchase amount, course titles, and copies sold will be printed.
   - Assertions will verify that the extracted values match the expected ones.