

Test 1, 4413-Lab01, Feb 5, 2020

How to proceed

1. Import the project from Lab 4 with the name YourStudentLastName_YourStudentNumber_Test1. Example: Lee_2109800_Test1. You will use this project for Test 1. From now on, we refer to the test project as Test1.
2. implement new specifications, as indicated in the Test instructions. You will see the test instructions in a Quiz, that opens after the your lab starts. There is only one question in that quiz, and, in the question body, you will have a link to the instructions.
3. When implementing the new specifications, make sure the functionality of Lab 4 remains unchanged, that is
 - if your application is accessed with <http://host:port/Test1/Osap>, or <http://host:port/Test1/admin>, it should exhibit the Lab 4 functionality (in other words, do not change any functionality of Lab 4)
4. **Do not add new servlets. You can add Java classes as needed.**
5. Make sure there are no compilation or SAX parser errors before exporting
6. Submit your work by
 - exporting the project to a war file and also making sure the *source files* are included in the export.
 - uploading YourStudentLastName_YourStudentNumber_Test1.war using Moodle Test1 page (again, make sure you exported the source). You can drag and drop the file or attach it. You have a button "Submit attempt" that you need to press before the quiz expire.
7. In the remainder of the text, "+" means a requirement
8. **No communication among the students or with external people or bots is allowed. No chat sessions, no email, no phone. You can use your material from class and Labs, including examples.**

I. Extend UI, Model, Osap

Make sure your Osap servlet can be accessed with "/Osap" and "Osap/*" where * means any string. Test this before going forward.

Your stakeholders would like an extended UI for your application so they collect information about applicants. Extend your application in this way:

+If a user uses this URL <http://host:port/Test1/Osap/Customizable>, the application should have these additional features

+The "legend" element of your calculator should include a copyright text. That copyright text is customizable in web.xml. Use your full name and student number as the copyright text in the web.xml. For example, Ma Li is the name of student, 12345 is the student number (remember, it has to be your personal data, taken from web.xml).

+When accessed with the above URL, the form should have a new input text field, "Applicant Number".

+Perform simple validation *at the server side*, on the Applicant Number. It should be greater than 0 and less than 10000. Validation results should be reflected in the UI. When a validation rule is violated, you need to display a meaningful error message in the user interface. Also,

prevent throwing validation exceptions in the server logs.

- +Persists the Applicant Number value among multiple requests in a session.

- +When accessed with the above URL, the form should have the "Grace Period" as an input, with numbers between 6 and 12.

- +Validate the input data above *at the client side* and display meaningful information in a pop-up window

- +When "Submit" button is clicked, the interest payment should be calculated with the grace period specified in the form (and not the fixed period from web.xml). **Ignore the "Submit (Ajax)" for this test.**

Calculator by Ma Li, 12345

Applicant Number*

Principal*
(total loan amount after studies)

Annual Interest Rate*

Payment Period*
(total number of months)

Grace Period *
(Take advantage of this grace period and include a
grace period interest with your loan balance)

Submit

Submit (Ajax)

- +The Result page shown for the above form should show the monthly payments and should include the Applicant Number and the Grace Period used for calculations. When press the Re-compute button, you should show the above UI.jspx, that is, the one you created during the test.

- +When invoked with <http://host:port/Test1/Osap>, your application should have the same UI, Result pages from Lab 4, and the same behavior as in Lab 4.

- +If the administrator uses <http://host:port/Test1/admin>, the application should return the Maximum Principal, as in Lab 4.

- +If the administrator uses <http://host:port/Test1/admin?p=applicantNumber>, the application should return the Maximum Principal and the Applicant Number who requested that principal

Notes

The test is 15% from your final grade

You will be marked for

- completeness and correctness of requirement implementation (a requirement is marked with + above): 80%
- robustness: 15% (includes validation error handling, 404 errors, etc..)
- coding style: 5%