

Total points: 80% of the total final project grade.

CS362-004

Final Project: Part-B

The final project is designed to check your cumulative understanding.

Notes:

- Use the URLValidatorInCorrect folder for Part-B.
- Submit one copy per group to Canvas (pdf) and the GitHub class repository (code).
- Add a comment in Canvas and give the URL of the student fork who submitted your project in GitHub (under final project).
- Write your group members details in the pdf files
- The code and documentation for the Apache Commons Validator is available in this link <http://commons.apache.org/proper/commons-validator/>
- The GitHub account is available at this link <https://github.com/apache/commons-validator>

Part-B, Testing URL Validator:

You are provided a buggy version of URLValidator and you will need to find as many bugs as you can. In Part-A, you were provided the current test framework that Apache commons team used to test URLValidator but you need to assume that those tests don't exist.

For this project, assume your team is a testing company and a client comes to you with a URLValidator implementation and asks for your help making it bug-free. You will only need to concentrate on isValid() method. Your task is to find the bugs, then find the failure causes and write up your findings.

Note: You don't need to fix any of the bugs. Developers will do it.

Use the various methodologies that you've learned to perform your testing:

1. Do manual testing. Call the isValid() method of URLValidator with different possible valid/invalid inputs. Document any failures that you find. (15 points)
2. Do programming based testing. Write several unit test cases that implement various methods to test different URLs. Document any failures that you find. (20 points)
3. Do random testing. Come up with good input partitioning. Try to provide a varying set of inputs that partition the overall input set well. Document any failures that you find. (15 points)
4. Submit a report called ProjectPartB.pdf in Canvas that contains the following Sections: (40 points)
 - Methodology Testing (15 points)

- Write in detail about each methodology you used for testing (manual, partition and programming based).
 - Write in detail the test functions that implement each methodology.
 - For manual testing, provide your URLs.
 - For random testing, describe your partitions and list some of the URLs that represent each partition.
- Bug Report (10 points)
 - Write a bug report for each of the bugs you found.
 - Describe the failure.
 - Describe how you found it terms of the test case that detected it.
 - Describe the cause of the failure. Explain what part of the code is causing it and provide a screenshot of the faulty code.
- Debugging (10 points)
 - When you find a failure, debug it using Eclipse/IntelliJ debugger or any other tool and find its cause.
 - Did you use any of Agan's principle in debugging URLValidator?
 - Describe the failure and your debugging details for each bug. Provide the line numbers and file names where the failure manifested itself.
- Team Work (5 points)
 - Write about how worked as a team? How did you divide your work? How did you collaborate? List the contributions of each of your team members.
- 5. The class GitHub repository (10 points)
 - Submit your unit tests/random tests on GitHub under projects/your-onid/URLValidatorInCorrect/ folder.
 - Create a new branch of your repository called "youronid-finalproject" that contains your final submission. This branch must be created before the due date to receive credit.

General Notes:

- Only one person needs to submit Part-b
- Write your names and onids (all group members) in the Canvas comment and the pdf files.
- Add a comment in Canvas and give the URL of the student fork who submitted your project in GitHub (under final project). (-10 for not submitting the URL)