Course Marking Policy

Typical Lecture Breakdown

- **6:05 - 6:20** (15 min): Recap of previous material
- **6:20 - 6:40** (20 min): Handwritten quiz
- **6:40 - 7:05** (25 min): Lecture
- **7:05 - 7:15** (10 min): Break
- **7:15 - 7:45** (30 min): Lecture
- **7:45 - 8:45** (60 min): Lab & code review time

Assignment Submission Process

Each assignment is completed online through GitHub Classroom.

General submission steps

- Access GitHub assignment link via Blackboard.
- Accept assignment and get your private repository created at GitHub.
- Copy the link to your repo.
- Create a local directory for all your assignments, e.g. 'security_assignments'.
- Clone assignment repo to your machine: `cd security_assignments` and `git clone [link]`
- Create your own branch, e.g. 'solution': `git checkout -b solution`
- Open cloned repository in your editor, e.g. VSCode (typically in `security_assignments/[cloned-repo-name]`)
- (Optional) Install required libraries for testing: `npm i`
- Work on your assignment, add your code into `exercise/` folder only.
- (Optional) Run automated tests locally to verify that your solution works: `npx cypress run`
- Commit your changes: `git add .` and `git commit -m "changes"`
- Push your code to GitHub: `git push -u origin solution`
- Create a Pull request (PR) between your branch and `main` branch.

- Wait until automated tests run to check your code in GitHub.
- If the tests pass successfully, merge your PR.
- Submit the link to your PR through Blackboard as a submission attempt.

Tips for Success

- Add descriptive PR title and detailed PR description. Clearly state what was changed and why. Example:

  PR Title: Complete memory leak task in Lab 3

  PR Description: This PR fixes memory leak in the assignment file caused by function running too many times. Solution was to clear result array at the beginning of each iteration of the loop.

- When adding your code, split your work into small meaningful git commits (at least 3). Example of commits:
  - Create html file for the solution
  - Add styles
  - Add JS code to trigger welcome message on button click
- Be prepared to write handwritten quiz on this material during next lecture.
- Be prepared if needed to meet with the instructor to demonstrate code understanding.

Code Review

Code Review will typically happen after you submitted your lab and quiz and suggested mark is already determined.

Code Review conditions

You would need to go through Code Review process if any of the following conditions occur:

- You were not present in the lecture
- Suspicious code patterns are detected in your submission (lab or quiz)

Code Review process

Meet with the instructor for a short code review session (typically up to 3 min) during your next lecture.

You will be asked about 3 questions related to your lab or quiz

Typical questions:

- Explain what is `innerHTML` property?
- Explain what `return false` is for?

You will have about 3 minutes to demonstrate your understanding.

After evaluation, the instructor will either:

- Confirm your understanding and apply the suggested mark, or
- State the level of your understanding and apply penalty (10% or 25%) for Poor Code understanding, or
- State that you failed to demonstrate understanding, in which case a grade of 0 stands

Suggestion: If you received "code review", thoroughly review your lab code and be prepared to answer any questions about it.

Marking Process

Your final mark is calculated through three sequential steps:

1. 50% of the mark is Technical Evaluation: Create a PR with the solution.
2. 50% of the mark is Understanding Evaluation: a 15-minute handwritten quiz on week material.
3. Deductions: Late submissions, Poor Code understanding or Low Code Quality.

Technical Evaluation (50%)

Automated tests along with manual checks by instructor determine if your code performs as required.

If all tests pass, you will receive 50% of the final mark. Otherwise, your score will reflect the percentage of tests passed. For example, if only 85% of the tests pass, you will receive 50% * 0.85 = 42.5% for the Technical Evaluation.

Understanding Evaluation (50%)

At the beginning of each next lecture you will have a 15-minute handwritten quiz (3-4 questions) on the previous week’s material and lab. No electronic devices are permitted.

You will need to either write a function or explain the code given to you.

Typical questions:

- Create a function in JS.
- Write one unit test.

Successfully completing the quiz will add another 50% of the final mark for the assignment. Quiz will be graded according to a rubric.

Deductions (if applicable)

- Late submission: deduction 5% per day (max 25%).
- JavaScript Code Quality Issues (ESLint): deduction 5%.
- Poor Code Understanding (based on conversation with the instructor): deduction 10% or 25%.

Examples

Example 1

Full mark for a lab is 3 marks. 30 out of 33 tests on Technical Evaluation step passed. The lab was submitted 1 day late. 8 out of 12 points earned on handwritten Quiz. No Code Quality issues. No code review with the instructor.

Final mark will be 2.2

Criteria	Calculation	Result
Full mark available		3
Technical Evaluation	30/33 tests passed * 50%	45.45%
Late submission deduction	1 day late	5%
ESLint Code Quality Check	ESLint check passed	0%
Understanding Evaluation (Quiz)	8/12 points * 50%	33.33%
Poor Code Understanding (code review)	N/A	0%
Total Percent Received	45.45% (Technical) + 33.33% (Understanding) - 5% (Late) - 0% (ESLint) - 0% (code review)	73.79%

Criteria	Calculation	Result
Final mark	3 * 73.79%	**2.2**

Example 2

Full mark for a lab is 6 marks. All tests on Technical Evaluation step passed. The lab was submitted on time. 3 out of 10 points earned on handwritten Quiz. No Code Quality issues. No code review with the instructor.

Final mark will be 3.9

Criteria	Calculation	Result
Full mark available		6
Technical Evaluation	All tests passed	50%
Late submission deduction	submitted on time	0%
ESLint Code Quality Check	passed	0%
Understanding Evaluation (Quiz)	3/10 points * 50%	15%
Poor Code Understanding (code review)	N/A	0%
Total Percent Received	50% (Technical) + 15% (Understanding) - 0% (Late) - 0% (ESLint) - 0% (code review)	65%
Final mark	6 * 65%	**3.9**

Example 3

Full mark for a lab is 10 marks. All tests on Technical Evaluation step passed. The lab was submitted on time. All points earned on handwritten Quiz. No Code Quality issues. Code review with the instructor was required.

Final mark will be 7.5

Criteria	Calculation	Result
Full mark available		10
Technical Evaluation	All tests passed	50%
Late submission deduction	submitted on time	0%
ESLint Code Quality Check	passed	0%
Understanding Evaluation (Quiz)	All questions answered	50%
Poor Code Understanding (code review)	Student didn't demonstrate sufficient code understanding	25%
Total Percent Received	50% (Technical) + 50% (Understanding) - 0% (Late) - 0% (ESLint) - 25% (code review)	75%
Final mark	10 * 75%	**7.5**

Plagiarism Policy

- If code contains **more than 8 identical lines** or **50 code tokens** to another student's work, you will receive **0** for the assignment and the coordinator will be notified.
- You will have **1 week** to meet with your instructor and explain why this has happened.
- If explanation is convincing, **maximum 50%** on resubmission.
- If unconvincing, the **0 stands**.

Extensions

Extensions may be granted for valid reasons (e.g., illness, emergencies) if requested **within 48 hours** of the assignment start. Documentation may be required.

Attendance

To be marked present, you must attend the entire lecture.

Attendance is taken in the beginning of each lecture.

Submission window

Assignments will be opened on GitHub Classroom immediately after the lecture concludes.

Typically, you have 2 days to submit your lab.

The latest chance to submit (if past due date with penalties) is 2 weeks from the assignment opening.

After the two-week period, any unsubmitted labs will automatically be marked as 0 without a chance to submit.

Automatic Zero conditions

Your lab will receive a grade of 0 if any of the following conditions occur:

- No Pull Request (PR) is created (code pushed directly to main branch)
- Files outside the designated exercise folder are modified