



Home / Skeleton Rubric for Assignments

Skeleton Rubric for Assignments

On this page

In code walks, your assignment mark is determined using a rubric. Each assignment has its own specialized rubric, but they are all derived from the skeleton rubric described here, and will look rather similar. Depending on the assignment and whether it is a code-walk on distinction-level content or not, both parts of the rubric and their weights may change.

Skeleton Rubric

COMP1110/6710, S2 2025

Time on Task Q1 _____ Q2 _____ Q3 _____ Q4 _____



Student Name: _____

Student UID: _____

Points: ___/6 Style +

___/14 Design +

___/20 Presentation =

___/40 Code Walk (min 0)

___/20 Tests +

___/40 Code Walk +

___/40 Tests * Code Walk / 20 -

___/50 Form Penalties =

___/100 * (1 if So0 correctly submitted, signed, 0 if not) =

___/100 Total

This percentage will be multiplied with the value of the assignment, which may depend on whether the code walk was for distinction-level content or not.

Section I: STYLE [___/6 points, subtractive, min -2]

=====

Deductions for:

- [-2] Messy indentation
- [-1] Other non-idiomatic formatting
- [-2] Function names changed from assignment
- [-1] Non-descriptive variable or function names ("lame names", repeatable, max 3)
- [-1] Tests without explanations (can use comments or error message)
- [-1] Messy comments (contracts, data definitions, etc.)
- [-1] Magic constants
- [-2] Using BigBang or Start when not needed/required for the problem
- [-1] Anything especially weird



Section II: DESIGN [___/14 points, min -3]

=====

The interviewer says:

"Please explain how you designed the function [function-name-here]. Follow the design recipe."

II.A Data Definitions/Templates [__/4 points, subtractive, min -2]


Deductions for:

- [-2] Each missing necessary data definition
- [-2] Each poorly designed data definition
(captures just enough information; all combinations are meaningful or else documented by an invariant; easy to understand and distinguish cases)
- [-1] Each field missing an interpretation that the grader judges to be necessary
- [-2] Each missing or wildly incorrect template
- [-1] Each flawed template

II.B Functions Follow the Recipe ("Free Fire Zone") [__/5 points, subtractive, min -5]

Start with the function to be codewalked, and check any related functions that you see.

Functions must have purpose statement, signature, examples, strategy, implementations, and tests.

Purpose must be clear, correct, and add at least some information beyond the signature. Function must match template/strategy, produce correct answer if its helpers do. Functions that could be written as simple instances of map, fold, filter are written that way, where allowed. 

Deductions (all these repeatable for each occurrence)

- [-1] Missing or incorrect signature, purpose statement, examples, strategy, or no tests (repeatable for each missing element)
- [-1] Inadequate purpose statement
- [-2] Function does not match template/strategy
- [-1] Excessively complex or repeated code that should have been put in a helper function
- [-2] Design that the grader judges to be overly complex.
Criterion: the grader can't follow the logic based on the purpose statements
- [-2] Function could be replaced by map, foldr, filter or some easy combination, where allowed (repeatable).

II.C Quality of Tests [___/5 points max]


[A list of required aspects of your tests with associated marks]

Make sure the students know how their test shows what it's supposed to show. If there is a test but the student doesn't know it, take off points under presentation

Section III: PRESENTATION [___/20 points, min 0]

=====

Presentation score includes all aspects of the presentation, not just answering the given questions. Issues such as not being able to explain code, find tests, give a concise answer or speak at an understandable volume are all fair game. A code walk should go through about 3-5 "major" components of an assignment. This number is question-dependent.

- Design recipe [___/2 points, subtractive, min -1]
[-1] needs prompting to follow design recipe (repeatable)
- Answering small questions [___/9 points, mixed, min 0, default +1]
[+1] for each speedy, concise, and correct answer to a non-trivial question
[-1/-2] for each question not answered satisfactorily (repeatable)
[-3] for each "major" component that was not covered due to taking too long to answer questions (repeatable) 
- General understanding [___/4 points, mixed, min -5, max 6] [Core/6/4]
[6/4] Perfectly understands everything, no notes, on track to HD
[4/2] Consistently solid understanding of course material, on track to HD
[2/0] Some flubs, but generally okay understanding, on track to HD
[0/-2] Several major issues in answering questions/understanding
[-2/-5] Severe lack of understanding
- Answering a question about course content [___/5 points]
[5] fully correct and concise answer
[4] slightly flawed answer
[3] flawed but largely correct answer
[2] answer contains a correct statement related to question
[1] answer contains a correct word related to question

We generally expect that a student will be asked lots (not just 8)

of smaller questions during a code walk. Satisfactory answers are concise and to the point. Wasting time incurs progress penalties.

Section IV: FORM PENALTIES [____/50 points]

=====

- [20] Code did not compile, but was fixed (no guarantee)
- [15] Test interface mismatch (also in addition to code did not compile if problems remain after fix)
- [10] Incorrect file names; must be: notebook.yml and as described in assignment
- [15] Insufficient git commits/pushes (must have at least one git commit+push per hour of work logged in notebook, at least one per work session, and at least one per assignment part)
- [15] Missing/severely broken notebook
- [5] Use of disallowed Java feature (repeatable per feature, according to assignment specifications)
- [5] Uninformative commit message or notebook entry (repeatable, max 15)
- [1] Absent from part of code walk (repeatable per minute, max 50)



Acknowledgement of Country

The Australian National University acknowledges, celebrates and pays our respects to the Ngunnawal and Ngambri people of the Canberra region and to all First Nations Australians on whose traditional lands we meet and work, and whose cultures are among the oldest continuing cultures in human history.

Contact ANU | Copyright | Disclaimer | Privacy |
Freedom of Information

+61 2 6125 5111 |
The Australian National University, Canberra

TEQSA Provider ID: PRV12002 (Australian University) |
CRICOS Provider Code: 00120C | ABN: 52 234 063 906

