FIT1008 / FIT2085 – Introduction to Computer Science – Assessment 2 – Marking Guide / Rubric

CRITERION	PERFORMANCE				
	POOR	UNSATISFACTORY	SATISFACTORY	GOOD	EXCELLENT
Task 1 and 2 – 5 marks total					
Quality Example Mistakes: *Instance variables used when class variables should have been or vice versa; multiple pre-conditions not enforced; multiple methods implemented in the Abstract Class when it should have been defined as abstract and vice versa; Abstract class not set-up correctly as ABC, precondition(s) not checked in abstract class if possible. Instance variables used when class variables should have been or vice versa; multiple pre-conditions not enforced; Premature exit due to failure*	Many mistakes / extremely low quality. (0 marks)	-	At most 1 mistake. Code quality is passable. (0.5 marks)	-	No mistakes. Great quality overall. (1 mark)
Documentation Example Major Mistakes: *No file header documentation; multiple methods missing type hints or type hints incorrect; complexity documentation absent on any function required, multiple methods with complexity documentation present but incorrect/incomplete; multiple methods missing method summarisation in function header.* Example Minor Mistakes: *One function missing type hints or type hints incorrect; one function with complexity documentation present but incorrect/incomplete; preconditions not documented in appropriate methods (each one missing is a separate error); single method missing method summarisation in function header.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (0.5 marks)	-	At most 1 minor mistake. Great quality overall. (1 mark)
Testing Fatal Mistake(all marks 0): *Not adding any test cases.* Example Major Mistakes: *Any test cases missing the requested descriptive comment; multiple test case descriptions not descriptive enough.* Example Minor Mistakes: *Single test case description not descriptive enough, not following given format for tests.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (0.5 marks)	-	At most 1 minor mistake. Great quality overall. (1 mark)

Correctness Example of common mistakes: *No pre-condition enforcement when invalid input given to function; not using the correct formula for attack damage or defence methods; not correctly defining __str__; Pokemon not instantiated correctly; Pokemon don't faint when HP goes to 0 (or below).*	Unable to pass any test cases (0 marks)	Able to pass basic test cases (0.5 marks)	Able to pass basic and edge test cases (1 mark)	Fails at least one test input given by the instructor (1.5 marks)	Passes all tests as given by the instructor (2 marks)
Task 3 – 15 marks					
Quality Example Major Mistakes: *Having significant repeated or convoluted code; battle modes not set up correctly, ADT selection is poor, the AI does not function as required, random team generation causes failures, program crashes while running; user facing methods don't handle invalid input if provided.* Example Minor Mistakes: *Having mildly repeated or convoluted code; mild issues with preconditions.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (1 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (2 marks)	No major errors and at most a few minor mistakes. Good code quality overall. (3.5 marks)	At most 1 minor mistake. Great quality overall. (5 marks)
Documentation Example Major Mistakes: *Insufficient commenting for multiple methods; multiple methods missing type hints or type hints incorrect; complexity documentation absent on any function required, multiple methods with complexity documentation present but incorrect/incomplete; multiple methods missing method summarisation in function header; no file header documentation.* Example Minor Mistakes: *One function missing type hints or type hints incorrect; one function with complexity documentation present but incorrect/incomplete; pre-conditions not documented in appropriate methods (each one missing is a separate error); single method missing method summarisation in function header; class constants not uppercase.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (2 marks)	At most 1 minor mistake. Great quality overall. (3 marks)
Testing Fatal Mistake: *Not adding any test cases.* Example Major Mistakes: *Any test cases missing the requested descriptive comment; multiple test case descriptions not descriptive enough.* Example Minor Mistakes: *Single test case description not descriptive enough, not following given format for tests.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (1 marks)	-	At most 1 minor mistake. Great quality overall. (2 mark)
Correctness	Unable to pass any	Able to pass basic	Able to pass basic	Fails at least one test	Passes all tests as

Example of common mistakes: *Teams not set up correctly, incorrect ADT choice, Pokemon not added to the team correctly, input validation not performed*	test cases (0 marks)	test cases (1 marks)	and edge test cases (2 marks)	input given by the instructor (3.5 marks)	given by the instructor (5 marks)
Task 4 – 10 marks					
Quality Example Major Mistakes: *Incorrectly followed battle logic, not returning pokemon to the right location, logic not fully translated correctly* Example Minor Mistakes: *Mild issues with preconditions; other small mistakes in implementation.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1.5 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (2.5 marks)	At most 1 minor mistake. Great quality overall. (4 marks)
Documentation Example Major Mistakes: *Insufficient commenting for multiple methods; multiple methods missing type hints or type hints incorrect; complexity documentation absent on any function required, multiple methods with complexity documentation present but incorrect/incomplete; multiple methods missing method summarisation in function header.* Example Minor Mistakes: *One function missing type hints or type hints incorrect; one function with complexity documentation present but incorrect/incomplete; pre-conditions not documented in appropriate methods (each one missing is a separate error); single method missing method summarisation in function header; class constants not uppercase.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (1.5 marks)	At most 1 minor mistake. Great quality overall. (2 marks)
Testing Fatal Mistake: *Not adding any test cases.* Example Major Mistakes: *Any test cases missing the requested descriptive comment; multiple test case descriptions not descriptive enough.* Example Minor Mistakes: *Single test case description not descriptive enough, not following given format for tests.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (1 marks)	-	At most 1 minor mistake. Great quality overall. (2 mark)
Correctness Example of common mistakes: *Not decreasing one hp when both pokemon are alive, returning a pokemon that is not alive to*	Unable to pass any test cases (0 marks)	Able to pass basic test cases (0.5 marks)	Able to pass basic and edge test cases (1 mark)	Fails at least one test input given by the instructor (1.5 marks)	Passes all tests as given by the instructor (2 marks)

the team, not following the battle mode order; battle_mode not using the correct ADT; battle method returns wrong value; pokemon battling from different positions.					
Task 5 – 10 marks					
Quality Example Major Mistakes: *Battle tower not implemented correctly, Iterator incorrectly set up, duplicates not removed on request, trainers with 0 lives not removed from the tower.* *Iterator does not stop iterating using the correct exception.* *Inefficient or poor choice of ADTs; user facing methods handle invalid input if provided (all units)* *Sorting not performed correctly (1054 only)* Example Minor Mistakes: *Mild issues with preconditions; other small mistakes in implementation.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1.5 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (2.5 marks)	At most 1 minor mistake. Great quality overall. (4 marks)
Documentation Example Major Mistakes: *Insufficient commenting for multiple methods; multiple methods missing type hints or type hints incorrect; complexity documentation absent on any function required, multiple methods with complexity documentation present but incorrect/incomplete; multiple methods missing method summarisation in function header.* Example Minor Mistakes: *One function missing type hints or type hints incorrect; one function with complexity documentation present but incorrect/incomplete; pre-conditions not documented in appropriate methods (each one missing is a separate error); single method missing method summarisation in function header; class constants not uppercase.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (1.5 marks)	At most 1 minor mistake. Great quality overall. (2 marks)
Testing Fatal Mistake: *Not adding any test cases.* Example Major Mistakes: *Any test cases missing the requested descriptive comment; multiple test case descriptions not descriptive enough.* Example Minor Mistakes: *Single test case description not descriptive enough, not following given format for tests.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (1 marks)	-	At most 1 minor mistake. Great quality overall. (2 mark)
Correctness	Unable to pass any test cases	Able to pass basic test cases	Able to pass basic and edge test cases	Fails at least one test input given by the	Passes all tests as given by the

Example of common mistakes: *Usually the same mistakes as quality but logical errors in your battle tower implementation could cause the test cases to fail, translation of attributes could also be incorrect*	(0 marks)	(0.5 marks)	(1 mark)	instructor (1.5 marks)	instructor (2 marks)
Task 6 – 10 marks					
Quality Example Major Mistakes: *Tournament creation is invalid, the instructions have most likely not been followed. The stages of the tournament have not been set up correctly / done in the incorrect order, incorrect/inefficient ADT used to setting up tournament; user facing methods handle invalid input if provided* *(all units)* *Tournament balanced check is faulty (1054 only)* Example Minor Mistakes: *Minor issues with tournament validity checks, tournament validity not checked prior to starting the tournament.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1.5 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (2.5 marks)	At most 1 minor mistake. Great quality overall. (4 marks)
Documentation Example Major Mistakes: *No file header documentation; multiple methods missing type hints or type hints incorrect; complexity documentation absent on any function required, multiple methods with complexity documentation present but incorrect/incomplete; multiple methods missing method summarisation in function header.* Example Minor Mistakes: *One function missing type hints or type hints incorrect; one function with complexity documentation present but incorrect/incomplete; preconditions not documented in appropriate methods (each one missing is a separate error); single method missing method summarisation in function header.*	Many major mistakes / extremely low quality. (0 marks)	Multiple major errors, quality of code is poor (instructions likely not obeyed). (0.5 marks)	At most 1 major error or many minor mistakes. Code quality is passable. (1 mark)	No major errors and at most a few minor mistakes. Good code quality overall. (1.5 marks)	At most 1 minor mistake. Great quality overall. (2 marks)
Testing Fatal Mistake: *Not adding any test cases.* Example Major Mistakes: *Any test cases missing the requested descriptive comment; multiple test case descriptions not descriptive enough.* Example Minor Mistakes: *Single test case description not descriptive enough, not following given format for tests.*	Many major mistakes / extremely low quality. (0 marks)	-	At most 1 major error or many minor mistakes. Code quality is passable. (0.5 marks)	-	At most 1 minor mistake. Great quality overall. (1 mark)

Correctness Example of common mistakes: *Incorrect logic while running the tournament, validity is not checked, tournament swapping logic is incorrect, result from each battle not stored and retrieved correctly, battle logic is flawed*	Unable to pass any test cases (0 marks)	Able to pass basic test cases (0.5 marks)	Able to pass basic and edge test cases (1 mark)	Fails at least one test input given by the instructor (1.5 marks)	Passes all tests as given by the instructor (2 marks)
Peer Evaluation – Multiplier for Group Work min((Mark / 15) + 0.2, 1)					
Contribution (10 marks) (a weighted mean value (see footnote) will be computed across all the peers' reports) *Use this section to award marks to your peers on how they contributed to the group work. Peer contribution includes how the team member helped with coding in the assessment and what their overall code contribution was during the course of the assessment.*	**No Contribution:** The student did not contribute to the solution. (0 marks)	**Poor Contribution:** The student's contribution to the solution was minimal. (2 marks)	**Limited Contribution:** The student submitted an adequate contribution to the solution. (6 marks)	**Good Contribution:** The student went above the norm towards their contribution and helped peers. (8 marks)	**Great Contribution:** The student provided an exceptional standard in their contribution to the solution. (10 marks)
Communication (5 marks) (a weighted mean value (see footnote) will be computed across all the peers' reports) *Use this section to award marks to your peers on how communicative they were during the course of the assessment. Communication encompasses all communication, written, meetings as well as attentiveness during applied assessment sessions and commitment to deadlines.*	**No Communication**: The student was largely absent from all group meetings and had no communication. (0 marks)	**Poor Communication:** The student's communication to the team was poor, mostly unable to reach. (1 mark)	**Limited Communication:** The student did not attend some of the agreed-upon meetings without explanation or warning (3 marks)	**Good Communication:** The student was present during meetings and communicated well with the group, attending most meetings. (4 marks)	**Great Communication:** The student attended all meetings and was exceptional in upholding communication standards. (5 marks)
Interview – 50 marks					
Student Knowledge (40 marks - fractional) *The student's knowledge of the subject matter demonstrated in the interview. This section covers the marks obtained for being able to answer 5 questions about the subject matter of the assessment. The questions will be randomly selected from the bank of questions provided.*	***Please refer to the interview specific rubric sheet provided on Moodle for the grading of this section.***				
Explanation of Design Choices Made (10 marks) *The student's explanation on why particular choices were made during the development of the solution for the assessment. The student should be able to explain how they came up with the solution and what choices they would make given no constraints. This section also covers marks for being able to answer questions regarding these design choices and scenarios that the demonstrator asks during the interview.*	**No Confidence:** The student demonstrated no confidence while explaining the design choices. (0 marks)	**Low Confidence:** The student demonstrated low confidence, unable to explain the design choices. (2 marks)	**Slight Confidence:** The student demonstrated slight confidence in the design choices, vaguely explaining them. (6 marks)	**Moderate Confidence:** The student demonstrated above average confidence, answering any questions about particular scenarios. (8 marks)	**High Confidence:** The student had exceptional confidence in design choices, answering any questions the demonstrator asked. (10 marks)

Footnote on Peer Evaluation weighting

In order to recognise the increased workload that groups of 2 or 3 will experience, and to limit the effect a single group member may have on the grade, we will calculate the weighted average peer evaluation in the following way:

- If you are in a group of 4, we simply sum up the peer evaluation from the 3 other students and divide by 3.
- If you are in a group of 2/3, we generate an average including 2/1 fake students who always give you 15/15.

For example, if you are in a group of 3, and receive a peer evaluation grade of 5/15 and 8/15 from your peers, then your peer evaluation mark would be:

$$(5 + 8 + 15) / 3 = 28 / 3 = 9.33 / 15$$

Which would give you a group work multiplier of 9.33 / 15 + 0.2 = 0.82.