ID630151: Introduction to Algorithmic Problem Solving

Portfolio Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	Portfolio contains comprehensive & robust evidence on the following: • Opens & runs in Unity without file structure & code modification. • Specified assessment & advanced assessment tasks.	Application contains clear & detailed evidence of functionality on the following: • Opens & runs in Unity without file structure & code modification. • Specified assessment & advanced assessment tasks.	Application contains evidence on the following: • Opens & runs in Unity without file structure & code modification. • Specified assessment & advanced assessment tasks.	Application does not, or does not fully contain evidence on the following: • Opens & runs in Unity without file structure & code modification. • Specified assessment & advanced assessment tasks.
Code Elegance	Application code thoroughly demonstrates code elegance on the following: • Intermediate variables. • Idiomatic use of control flow, data structures & other in-built functions. • Efficient algorithmic approach. • Sufficient modularity. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • File header & in-line comments. • Formatted script files. • No dead or unused code.	Application code clearly demonstrates code elegance on the following: • Intermediate variables. • Idiomatic use of control flow, data structures & other in-built functions. • Efficient algorithmic approach. • Sufficient modularity. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • File header & in-line comments. • Formatted script files. • No dead or unused code.	Application code demonstrates code elegance on the following: • Intermediate variables. • Idiomatic use of control flow, data structures & other in-built functions. • Efficient algorithmic approach. • Sufficient modularity. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • File header & in-line comments. • Formatted script files. • No dead or unused code.	Application code does not or does not fully demonstrate code elegance on the following: • Intermediate variables. • Idiomatic use of control flow, data structures & other in-built functions. • Efficient algorithmic approach. • Sufficient modularity. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • File header & in-line comments. • Formatted script files. • No dead or unused code.

Documentation & Git Usage	README file contains comprehensive evidence of: • URLs to resources used to build your games. Git commit messages comprehensively formatted & reflect the feature changes in concise detail.	README file contains clear evidence of: • URLs to resources used to build your games. Git commit messages clearly formatted & reflect the feature changes in substantial detail.	README file contains evidence of: • URLs to resources used to build your games. Git commit messages formatted & reflect the feature changes in detail.	README file does not or does not fully contain evidence of: • URLs to resources used to build your games. Git commit messages do not or do not fully formatted & reflect the feature changes.

ID630151: Introduction to Algorithmic Problem Solving

Portfolio Assessment Marking Cover Sheet

Name:

Date:

Learner ID:

Assessor's Name:

Assessor's Signature:

Criteria	Out Of	Weighting	Final Result
Functionality	10	70	
Code Elegance	10	20	
Documentation & Git/GitHub Usage	10	10	
Final Result			/100
This assessment is worth 100% of the final mark for the Introduction to Algorithmic Problem Solving course.			

Feedback:

Functionality:

Code Elegance:

Documentation & Git Usage: