

## ID511001: Programming 2

### Project 1: Pong Marking Rubric

	10-9	8-7	6-5	4-0
Functionality	The Pong application contains comprehensive and robust evidence on the following functionality: No code/file structure modification, game driven by one timer, ball and paddle are created using the Graphics class, collision detection between the ball, paddle and screen, user control with up and down keys, scoring system, double buffering, high score system, sound, game states and random colours.	The Pong application contains clear and detailed evidence on the following functionality: No code/file structure modification, game driven by one timer, ball and paddle are created using the Graphics class, collision detection between the ball, paddle and screen, user control with up and down keys, scoring system, double buffering, high score system, sound, game states and random colours.	The Pong application contains evidence on the following functionality: No code/file structure modification, game driven by one timer, ball and paddle are created using the Graphics class, collision detection between the ball, paddle and screen, user control with up and down keys, scoring system, double buffering, high score system, sound, game states and random colours.	The Pong application does not or does not fully contain evidence on the following functionality: No code/file structure modification, game driven by one timer, ball and paddle are created using the Graphics class, collision detection between the ball, paddle and screen, user control with up and down keys, scoring system, double buffering, high score system, sound, game states and random colours.
Code Elegance	The Pong application demonstrates comprehensive evidence on the following: <ul style="list-style-type: none"> <li>• Use of OO principles, i.e., encapsulation, abstraction, inheritance and polymorphism.</li> <li>• Use of intermediate variables, constants and enumerations.</li> <li>• Idiomatic use of control flow, data structures and in-built functions.</li> <li>• Efficient algorithmic approach.</li> <li>• Sufficient modularity.</li> <li>• Commenting and formatting.</li> <li>• No dead or unused code.</li> </ul>	The Pong application demonstrates clear evidence on the following: <ul style="list-style-type: none"> <li>• Use of OO principles, i.e., encapsulation, abstraction, inheritance and polymorphism.</li> <li>• Use of intermediate variables, constants and enumerations.</li> <li>• Idiomatic use of control flow, data structures and in-built functions.</li> <li>• Efficient algorithmic approach.</li> <li>• Sufficient modularity.</li> <li>• Commenting and formatting.</li> <li>• No dead or unused code.</li> </ul>	The Pong application demonstrates evidence on the following: <ul style="list-style-type: none"> <li>• Use of OO principles, i.e., encapsulation, abstraction, inheritance and polymorphism.</li> <li>• Use of intermediate variables, constants and enumerations.</li> <li>• Idiomatic use of control flow, data structures and in-built functions.</li> <li>• Efficient algorithmic approach.</li> <li>• Sufficient modularity.</li> <li>• Commenting and formatting.</li> <li>• No dead or unused code.</li> </ul>	The Pong application does not or does not fully demonstrate evidence on the following: <ul style="list-style-type: none"> <li>• Use of OO principles, i.e., encapsulation, abstraction, inheritance and polymorphism.</li> <li>• Use of intermediate variables, constants and enumerations.</li> <li>• Idiomatic use of control flow, data structures and in-built functions.</li> <li>• Efficient algorithmic approach.</li> <li>• Sufficient modularity.</li> <li>• Commenting and formatting.</li> <li>• No dead or unused code.</li> </ul>

<b>Documentation &amp; Git Usage</b>	<p>README file contains comprehensive evidence on the following:</p> <ul style="list-style-type: none"><li>• The Pong application's UML diagram.</li><li>• References to used code snippets.</li><li>• Known bugs if applicable.</li></ul> <p>Git commit messages comprehensively reflect the changes in concise detail.</p>	<p>README file contains clear evidence of:</p> <ul style="list-style-type: none"><li>• The Pong application's UML diagram.</li><li>• References to used code snippets.</li><li>• Known bugs if applicable.</li></ul> <p>Git commit messages clearly reflect the changes in substantial detail.</p>	<p>README file contains evidence of:</p> <ul style="list-style-type: none"><li>• The Pong application's UML diagram.</li><li>• References to used code snippets.</li><li>• Known bugs if applicable.</li></ul> <p>Git commit messages reflect the changes in detail.</p>	<p>README file does not or does not fully contain evidence of:</p> <ul style="list-style-type: none"><li>• The Pong application's UML diagram.</li><li>• References to used code snippets.</li><li>• Known bugs if applicable.</li></ul> <p>Git commit messages do not or do not fully reflect the changes.</p>
--------------------------------------	--	--	--	---

## ID511001: Programming 2

### Project 1: Pong Marking Cover Sheet

Name:

Date:

Learner ID:

Assessor's Name:

Assessor's Signature:

Criteria	Out Of	Weighting	Final Result
Functionality	10	40	
Code Elegance	10	45	
Documentation & Git Usage	10	15	
Final Result			/100
This assessment is worth 25% of the final mark for the Programming 2 course.			

**Feedback:**

**Functionality:**

**Code Elegance:**

**Documentation & Git Usage:**