

## WeThinkCode\_ Student Progress Report

<b>Student Name:</b>	<b>Lunga Malinga</b>
<b>Student Username:</b>	<b>lumalin022</b>
<b>Cohort:</b>	<b>2022</b>
<b>Campus:</b>	<b>Johannesburg</b>
<b>Date:</b>	<b>7 June 2023</b>

### Performance Scale Descriptors

Performance Scale Descriptors					
Level	5	4	3	2	1
Outcomes	Exceeds Expectations	Above Expectations	Meets Expectations	Below Expectations	Expectations not Met

#### Interpreting the Outcomes

**Exceeds Expectations:** student is able to submit project requirements well in advance before the deadline. Grading was passed/correct on the first attempt.

**Above Expectations:** student attempted grading twice before the deadline, but passed on the second attempt before the deadline.

**Meets Expectations:** student is able to submit projects on time and pass at the required level.

**Below Expectations:** student is below expectations, attempted grading and submitted after the deadline, and passed.

**Expectations not Met:** student attempted to submit after the deadline passed/failed and this indicates that major development is required. Students should make use of the interventions available to improve performance.

# Performance Report

<b>Year 1 Module 2 - Object-Oriented Programming</b> This module is an introduction to software design through the lens of Object - Oriented Programming (OOP). In this module, Java is introduced as the programming language. The switch from Python to Java further entrenches the fundamentals of programming.		
Project Name	Is the student able to:	Level
Unit testing in java	Use JUnit to reinforce existing testing techniques and gain familiarity with the Java toolchain.	5
Encapsulation in Java	Create classes to implement behaviour, hide and govern access to data.	5
Inheritance Polymorphism Composition	Use abstraction as an instrument of design. Implement specialisation of objects at run-time, based on the abstraction designed. Combine objects together to express more complex objects.	5
Unit Testing Encapsulation Polymorphism Inheritance with abstract classes Composition with interfaces	Apply Java constructs to design a cohesive solution.	5
Group Project (Robot Worlds)	Network programming using sockets; concurrency using threads; implementing a simple application protocol; serialisation and deserialization of data for transmission over the network.	3.3

I hereby certify that these are the official results for Lunga Malinga for Year 1, Module 2.



**Lwazi Gumede**  
**Head of Performance**  
[lwazi@wethinkcode.co.za](mailto:lwazi@wethinkcode.co.za)