**Object Oriented Programming 2 AL\_KCNCM\_9\_1: 2024 – 25**

**Assignment**

A black and gold logo

Description automatically generated

Submission Date: 04/04/2025

Lecturer Name: Dr Sean Kennedy

**Student**

Philip Herweling

A00326153

A00326153@student.tus.ie

AL\_KCNCM\_9

Contents

[Introduction 3](#_Toc188894483)

[Project Overview 3](#_Toc188894484)

[User-Stories 3](#_Toc188894485)

[UML Diagrams 3](#_Toc188894486)

[Evaluation 3](#_Toc188894487)

[Conclusion 3](#_Toc188894488)

[References 3](#_Toc188894489)

# Introduction

For this assignment I decided to not refractor the assignment I don’t for object-oriented programming 1. I wanted to start with a fresh idea and from scratch as I believe it will make me a better Java programmer. In the sections below I will begin by giving a brief description of the project domain. I will then have the user-stories which will focus on the features rather than the project domain. I will also include a UML class diagram before I evaluate my assignment referring to the brief. Then finally I will give a conclusion.

# Project Overview

The fitness tracker allows users to log their workouts, set fitness goals, and track their progress over time. It includes features such as calorie tracking, workout categorisation, and progress reporting.

## User-Stories

1. Lambdas and Streams

As a user , I want the application to filter my workout log by workout type and then also calculate the total calories burned for each work out type. That way I can see how effective my cardio or strength workouts are.

The key Java features I will be using to implement this user story are:

* Filter() for the work out types
* mapToDouble and sum() for the calorie calculation

1. Switch Expressions

As a user I want to get updates on my progress based on my personal fitness goal. Using a switch expression will allow the system to provide specific feedback for my goal type.

The key java features I will be using to implement this user story are:

* Enhanced switch expressions
* Pattern matching for goal types

1. Sealed classes

As a developer I want to make sure that new goal types specified by the user are limited to predefined categories. I can do this by using sealed classes, so that the codebase remains maintainable and extensible.

The key java features I will be using to implement this user story are:

* Sealed classes/Interfaces fir fitness goals
* Strong control over the hierarchy

1. Date/Time API

As a user, I want to be able to see how many days have passed since I last done a workout, this way I can track my consistency in my fitness routine.

The key java features I will be using to implement this user story are:

* LocalDateTime for storing workout times
* Duration.between() for calculations

## UML Diagrams

# Evaluation

# Conclusion

# References