****

**OBJECT ORIENTED PRINCIPLES**

ASSIGNMENT 3: A JAVA APPLICATION FOR MANAGING A STORE.

**Declaration of Authorship**

I, Diarmuid Mahony declare that the work presented in this assignment titled ‘A Java Application for Managing a Store’ is my own. I confirm that:

* This work was done wholly by me as part of my BSc. (Hons) in Software Development, my Msc at Munster Technological University.
* Where I have consulted the published work and source code of others, this is always clearly attributed.
* Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this assignment source code and report is entirely my own work.

On \_11\_\_DATE\_11/12/2022

Signature:

Diarmuid Mahony

**Java Application Description.**

This Java application where it will implant an interface where a user can add a product and they can also

get the cost of the product. The functionality of the application includes:

Adding/removing users and the items of a grocery store( the cost of the product).

Displaying the information of both the user/item/cost.

**Technical Difficulty: OOP Concepts Demonstrated in the Java Application.**

Encapsulation.

The class person.java models a user of a grousery store and it will add the user to array by adding it to a fuction called newCoustomr.

Aggregation

The class Product.java has a private field Id number an object of the type id.

Inheritance

The classes Item.java and Groroery.java also buying.java inherit from Id.java also item inherit from Groserys.java.

Also, the classes OrderItem.java will inherit from Grorerys.java and then it will be added into the main java file.

Class Hierarchy

The classes person.java and Product.java will be hierarchy for the class id.java so all the classes will get the id for all the id information.

Static classes

The classes that I have made static is by using it in the main.java programming as I have put static data that will not be moving or changing their values in the programming as they are only for the use of just testing.

**UML Design: Java Application.**

**Diagram

Description automatically generated**

**Testing the Java Application.**

The functionality of the application is tested in MyMain.java where it is based on multiply ways of switch staments and where I have 6 total amount of application.

The most of the applition Is to test out what I wanted to get done where I am testing out all of the application by making total of testing such that I made it were I have a a application of testAddUsesr and more like that and for each switch clause to select which methods to try and run each for the Java application.