|  |  |
| --- | --- |
| Student ID | D10126532 |
| Student Name | John Warde |
| Student Email | [john.warde@gmail.com](mailto:john.warde@gmail.com)  [john.warde@mydit.ie](mailto:john.warde@mydit.ie) |
| Title | Continuous Assessment – Part 1 – Library System |

# Background

As I have never programmed in Java professionally, having only competed a training course for Sun Certified Java Programmer, I decided to use the console/command line for user interaction.

# Class and Interface Descriptions

TODO:

## Library Class

Stub class for application startup

## LibraryItem Class

Abstract class which Book, Dvd and Periodical concrete classes are derived from. Implements OutputStrategy interface to provide a consistent calling mechanism for display to the console – an example of the Strategy pattern

## Book Class

Inherits from LibraryItem parent class

## Periodical Class

TODO:

## Dvd Class

TODO:

## Catalog Class

Maintains a list of all library items. Implements the Traverser interface, an example of the Iterator pattern, needs a better name and one that doesn’t clash with Java’s Iterator interface.

## CatalogViewer Class

Views/sub-sets of a catalog can be created by this class i.e. for listing all items on loan

## LibraryRepository Class

This is a singleton class, it is the data store for all the elements of this library system i.e. users, library items, loans?

## LibraryItemFactory Class

This is used to help populate the initial library Catalog, it takes input from a Comma Seperated Values type structure and return one of the concrete child classes of LibraryItem

## Members Class

This is the “data store” for all users of the library it also uses the Traverser interface to iteration over all users.

## User Class

Contains an individual user’s details

## OutputStrategy Interface

TODO:

## Traverser Interface

Uses the iterator pattern to provide consistent access to Library and User objects.

## LibraryManagerConsole class

Manages the console application including menus, lists and forms

The following classes are used to manage the console UI elements such as Menus, due to time constraints; I cannot list out the full details of all these classes.

Form

Manages the input and validation of a form that can contain one or more fields

FormField

Field definition of a single user input item, it is the parent class

FormFieldDate

FormFieldInt

FormFieldString

Confirmation

Presents a confirmation message to the with a yes or no answer.

ListChoice

Presents a list of menu choices and asks the user to make a valid choice.

ListContextMenu

This is used after a list is presented to the user and validates the user’s input based on the supplied count of listed items.

Menu

Presents a regular menu to the user and validates the user’s choice.

# Summary & Conclusions

TODO: