FIT1051 Assessment Four

Submission deadline: Thursday 6th June 11:55pm AEST (11:55pm MYT) via Moodle

Weight: This assessment is worth 38% of the unit total. Submission of your code is worth 50% of this total, and your interview component (held online in the following week) is worth 50%.

Late Penalty: 10% mark deduction per day

Instructions: Below are the coding tasks that you need to complete for Assessment 4. Your work and your submission should be independent. Please download the IntelliJ project folder below and unzip it. This will provide you with partial code in which to program your answers. Please complete each task in the appropriate section of the partial code, and submit to Moodle upon completion.

Academic Integrity: Please be reminded of the academic integrity standards that are expected of you at Monash, which were mentioned in Week 1. You should code alone and ask the unit staff for help if needed. Do not post your code in public forums or send your code to anyone. Do not copy/paste code from other sources and present it as your own — this includes use of generative AI tools. Breaching these academic integrity requirements can incur serious penalties.

Scenario: You are required to develop a management system for a music store that keeps track of stock and customers. It sells records and band t-shirts. The program should maintain a list of items that are in store at any one time, as well as customers and their orders.

When first run, it should present the user with a menu that has the following options:

1. Add a new customer.
2. Display a list of all customers.
3. Add a new record to the store inventory.
4. Add a new t-shirt to the store inventory.
5. Display a list of all items and their current stock levels.
6. Update the stock level of an item in the inventory
7. Enter and store the details of a new customer order. The order must belong to an existing store customer, and must be for an item that is in stock.
8. Output a list of all orders placed in the current month.
9. Exit the program.

This menu should continue running in a loop until Option 9 is selected by the user.

The **store** holds the following details about each **customer**:

- A unique **customer number**
- The customer's **name**
- A collection of all **purchases** that the customer has made

The store holds the following details about each item for sale:

- A unique **item number**
- **Artist/band name**
- **Price**
- Number of **items in stock** (can be 0 or more)

Note that records also have the **name of the album**, while t-shirts have **a size** (small, medium, large, or x-large).

An **order's** details consists of:

- A unique **order I**D
- A reference to the **one or more items** that have been bought in that order
- The **date of purchase**
- The purchase **total**

For full marks, your design should contain all of the following:

- **Inheritance** and the use of an **Abstract** class.
- An **appropriate interface** (Lesson 12.3) which ensures that can allow easy running of all required functionality.
- At least one **enum** type, used for an appropriate purpose (you can use more than one if you like, but you are only required to use one).
- **At least six classes** overall (including any abstract classes, and the Date class, if used).

Code marking rubric: Code design and correctness (50 marks total)

1. Program functionality (25 marks)

- Implementing overall main menu structure correctly
- Adding a new customer correctly to the store
- Displaying all customers
- Adding a new record to stock
- Adding a new t-shirt to stock
- Displaying all items (using appropriate calls to toString methods)
- Updating the stock level of an item
- Adding a new customer order
- Displaying the recent order details to the screen (using appropriate calls to toString methods) as described

2. Design requirements (20 marks)

- Sensible overall class design and adherence to good design principles
- Modularisation of complex tasks within classes
- Validation checks on user input
- Appropriate and correct use of inheritance (including abstract)
- Appropriate and correct use of enum type

3. Adherence to coding standards (5 marks)

- All code written should adhere to the guidelines set out in the FIT1051 Coding Standards.

Suggestions for Implementation: This is the biggest program you will have written for the unit! Make sure you approach it in a way that lets you tackle it bit by bit (i.e. problem decomposition). Suggested strategies include:

- Coding classes that do not rely on any other class first, and doing simple testing of them.
- Coding a simple menu and being able to make that work (and exit appropriately).
- Considering how menu items align with methods you write (and again how you break up the problem) and writing those, one menu item at a time, in order of complexity.

Do it bit by bit, breaking it up into logical chunks. And test... test... TEST!

Submission Instructions: Please submit your IntelliJ project folder as a .zip file and submit to the Assessment 3 link on the Moodle Assessments page as shown below. If you are not sure how to zip your project, please refer to the video here. MAKE SURE YOU DOWNLOAD FROM MOODLE AFTER THAT TO CHECK IT IS THE RIGHT SUBMISSION!

Interview component: You will be asked to demonstrate your program at an interview in Week 15, following the code submission date. You will be contacted by EMAIL to book an online interview time.

The interview will be up to 15 minutes in length, and consist of 6 questions. You may be asked to explain your code, your program designs, to modify your code, to discuss your coding decisions, or to explain the any of the coding concepts taught in Weeks 1-12 that this Assessment covers.

Interviews will take place online via Zoom. You must have access to a stable internet connection and a working webcam, and your webcam must be switched on for the duration of the interview. Interviews will be recorded for marking integrity purposes, and recordings will be deleted at the end of semester.

It is your responsibility to make yourself available for an interview time during Week 15, and to attend your online interview on time. The interview is worth 50% of your assessment mark, and any student who does not attend an interview will receive a fail grade for the assignment. Your interview must take place before the end of Week 15.

If you cannot attend your interview you MUST apply for Special Consideration.