**SCHOOL OF DIGITAL MEDIA AND INFOCOMM TECHNOLOGY**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**ST1011 APPLICATION DEVELOPMENT**

**ASSIGNMENT 1 (2015/2016 S2)**

**Objective of Assignment**

To allow students to practice what they have learnt in the module by developing an electronic point-of-sales (POS) system.

**Instructions and Guidelines**

1. The assignment consists of 2 deliverables. The assignment will account for 25% of your final grade.

2. The first deliverable should be submitted in class in Week 5: **16-Nov-2015 to 20-Nov-2015** to your tutor.

3. The second deliverable should be submitted by **03-Jan-2016 (Sun), 11.59 pm**.

You are required to submit a softcopy of your C# project (in zip file format) on BlackBoard. Your Visual Studio solution containing all code and data files (if any) must be submitted in a .zip file via the assignment drop box on ST1011 Blackboard site.

4. The first deliverable will account for 10% of your assignment grade.

5. The second final deliverable will account for 90% of your assignment grade.

6. You are encouraged to work in a pair. Marks will be given separately for each student in the team, depending on his/her contribution to the assignment.

7. The interview for the assignment will be conducted in week-12: **04-Jan-16 to 08-Jan-16**. You may be asked to explain the program and its logic and modify the program during the interview.

8. No marks will be awarded, if the work is copied or you have allowed others to copy your work.

9. **5 marks** will be deducted for each day of late submission, for up to 25 marks.

10. Any work submitted **more than 5 working days** after the deadline will not be accepted.

**Assignment Overview**

Design and implement a point-of-sales (POS) application to sell products of your **choice**. The system must support all operations (not limited to the following), for example:

1. Loading the product information from a text file.
2. Displaying a main menu for navigation.
3. Displaying summarised details (Title, Category, Price) of all products.
4. Displaying products belonging to a particular category. Your system should support a minimum of 2 categories.
5. Selecting products the user wants to purchase and storing it in memory. Allow user to check out.
6. Before checkout, allow the user to enter the secret member discount code to get the member discount. Users who don’t enter the code will not get the discount.
7. Display details of purchase such as title, category, cost of each product a user want to purchase and finally the total cost(after discount if any) when user confirms purchase and display the receipt.
8. Any other operations that fits your POS application.

The following are ***some*** suggested classes that you may use in the design and implementation of the POS application. All classes that you design must be used in your application, and encapsulate one or more data properties and methods. For each class that you write, you may decide to have as many or as few properties, but the property must support your application. Similarly, you may add methods to fit the needs of your application.

**The Product Class**

The **Product** class will be used in your POS application and represents the abstraction of product objects.

**The Category Class**

The **Category** class will be used in your POS application and represents the abstraction of category objects. A product may belong to **1 or more** categories.

**The Catalogue Class**

The **Catalogue** class will be used in your POS application to represent a collection of product objects. There are **many** products in a catalogue.

**The Cart Class**

The **Cart** class represents the products selected held by a user for purchase. Users may add to cart, remove from cart, check out the cart and generate a receipt. You will need to tailor the behaviour (functions) and data to the business rules of the POS system you are designing.

**The Transaction Class**

This class record the transaction datetime, the total amount for the purchase and the quantity of each product purchased.

**Other Classes**

Design and write any other classes and methods that will facilitate the testing and debugging of your POS system. For example, you may need methods/classes to handle the member discounts or users.

**Technical Requirements of First Deliverable (10%)**

You should submit the following to your tutor in Week 5 in the form of a 1-2 pages written proposal:

1. Objectives of your POS system. What products are you selling? What categories?
2. The list of candidate classes: class names, properties and methods.
3. App wireframe – the controls on the main form, layout of other forms and how the forms interact with each other. This may be hand-drawn.

**Technical Requirements of Second Deliverable (90%)**

Your assignment will be graded based on the following features:

1. Encapsulation using C# classes
   1. Use of properties and appropriate access modifiers
   2. Use of methods and appropriate access modifiers and parameters
2. Inheritance using C#
   1. Derive at least a class from 1 parent class
   2. Override at least 1 method of base class
   3. Call the constructor of the base class from a derived class
   4. Demonstrate method overloading with different method implementations for one method name
3. GUI
   1. Your program should comprise at least the following forms:
      1. - A splash screen
      2. - Main form (for the POS system)
      3. - A form to display the receipt after a purchase by the user.
   2. The POS system must be implemented using Windows Form controls.
   3. The POS system must be implemented using Windows Forms event handling.
4. The program must be able to track the total of each purchase transaction. The total should be clearly displayed.
5. BONUS features for Deliverable
   1. Allow customization of the GUI components such as allowing the user to select color themes.

**Submission**

1. Maximum team size is 2 persons per team. Each team member must contribute to the design and implementation of 1 or more classes. An individual multiplier may be applied for team members who do not contribute a substantial portion of work.
2. Your final submission should list down any **changes in written form** if you have made changes to classes and form design during development after Part 1 submission. You may update the submission to show the changes.
3. Take note that partial day late submissions will be taken as a day late.
4. Reduce the size of your zip files by deleting the /debug and /release folders.
5. **Only .zip files will be accepted. RAR files will not be accepted for submission.**
6. Ensure that your zip file contains **all required source code or data files**.
7. You are responsible to verify that the entire project can be built and run in both Debug and Release builds with Visual Studio. **Marks will be deducted if your solution fails to build and run during demonstration in class.**
8. You will be required to demonstrate your application as a team during the first practical session after the assignment due date. Team members who fail to be present during demonstration will have marks deducted.

**Warnings**

1. Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.