Bachelor of Information Technology – BInfoTech

Faculty of Business

IT5x84 – Programming

Assignment 2

Semester 1, 2017

Due Date : Week 11 (28 May 2017)
Total Marks : 25 marks

This is an individual assignment. Submit your work in an electronic format via Moodle site. Please place all files in a compressed folder prior to submission.

Academic Dishonesty

Plagiarism is the use of someone else's work without attributing it. Plagiarism is seen as academic dishonesty and is a serious and punishable academic offense. Award of a zero mark will be given for such offense.

Late Submission Penalty

If an assignment is handed in late without an extension granted in writing, a zero mark will be awarded.

Option 1

Develop a software application for the following scenario.

An automobile shop sells its own brand of lightbulbs. It keeps a table of equivalent part numbers to cross reference to major brands in a comma-separated values file. The shop wants to computerised the process of looking up part numbers in order to improve its customer service.

Shop	Brand A	Brand B	Brand C
LB123	KF156D	SD145G	6B45TY
LB124	BN416D	N9Y8WE	96Y6YB
LB125	GBYU15	MIOJCG	J1YT15
LB126	59HY8N	5264DEV	N5YTJH
LB127	RXU904	45TRVU	451NJ
LB211	H9V6KH	IUMHVE	5NJYT
LB212	6841ER	JIOTB4R	546NJY
LB213	5645TV	14568TR	5648KI
LB214	RTE435	8YU9T5	4265ER

The user should be able to enter the part number and brand, and look up the corresponding shop's part number. If writing a GUI application, you may allow the user to select the brand from a list or radio buttons.

Contents of the table above must be stored in a .csv file, and loaded into the application upon runtime. You can choose from two approaches for the look-up table. Store the part numbers either in a 2D array or in an array of structure. In either case, use the part number and brand entered by the user; look up and display the shop's part number. Allow the user to enter/append new data into the .csv file.

The features to include are:

1. Proper naming convention used
2. Meaningful internal documentation
3. Meet the functional requirements
4. Loads values from a .csv file
5. Correct search logics
6. Use of array or structure
7. Input verification and error checking
8. Suitable variables declaration and conversion
9. Codes are largely independent from interface objects
10. Additional properties setting in place, e.g. tab sequence, default accept/cancel button, input/output font alignment, etc.

Marks Allocation

Each feature gains 10 marks, for a total of 100 marks.

Option 2

The requirements:

1. You are required to create a snake game with the following features
2. The game should allow a snake to move around the screen with each timer tick
3. The game should end if the snake touches the walls or itself
4. Apples should be randomly generated and put on the screen
5. The snake should grow if it eats an apple
6. A score should be updated as apples are collected
7. When the snake dies the score should be saved to a file
8. A button to show the scores currently stored in the file is also required
9. A new game button is required to restart the game
10. Additional features you may think of

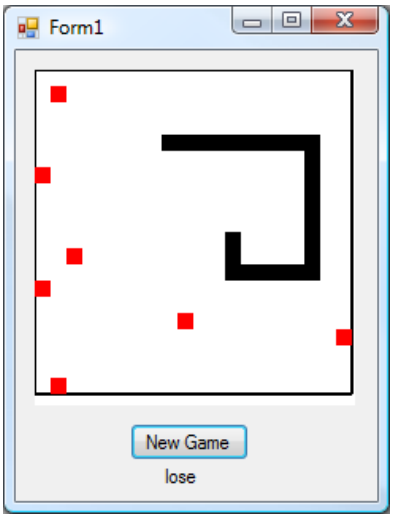
The tasks:

You are to create a snake game using a picturebox to display a snake and an arraylist to represent the snake.

This option requires a lot of thought about what you are doing. For guides on how to develop the application, please refer to the appendix. Try to follow the approach to construct (and test) the game in stages.

Marks Allocation

1. Forms design and Usability — 20 marks
   - Menu items
   - Buttons
   - Menu shortcuts
   - Message boxes where appropriate
2. Coding standards/structure — 35 marks
   - Variable and control names
   - Appropriate data types used for variables
   - Code comments
   - Error catching where appropriate
   - Quality (simple/readable/correct)
3. Requirements were met, including: — 45 marks
   - The snake moving
   - The snake growing
   - Apples on the screen
   - A score being kept
   - The score being written to a file and later read
   - A new button
   - The game ending when appropriate



-- THE END –