



Bachelor of Information Technology – BInfoTech

Faculty of Business

IT5x84 – Programming

Assignment 1

Semester 1, 2017

Due Date : Week 6 (8 April 2017)

Total Marks : 20 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.

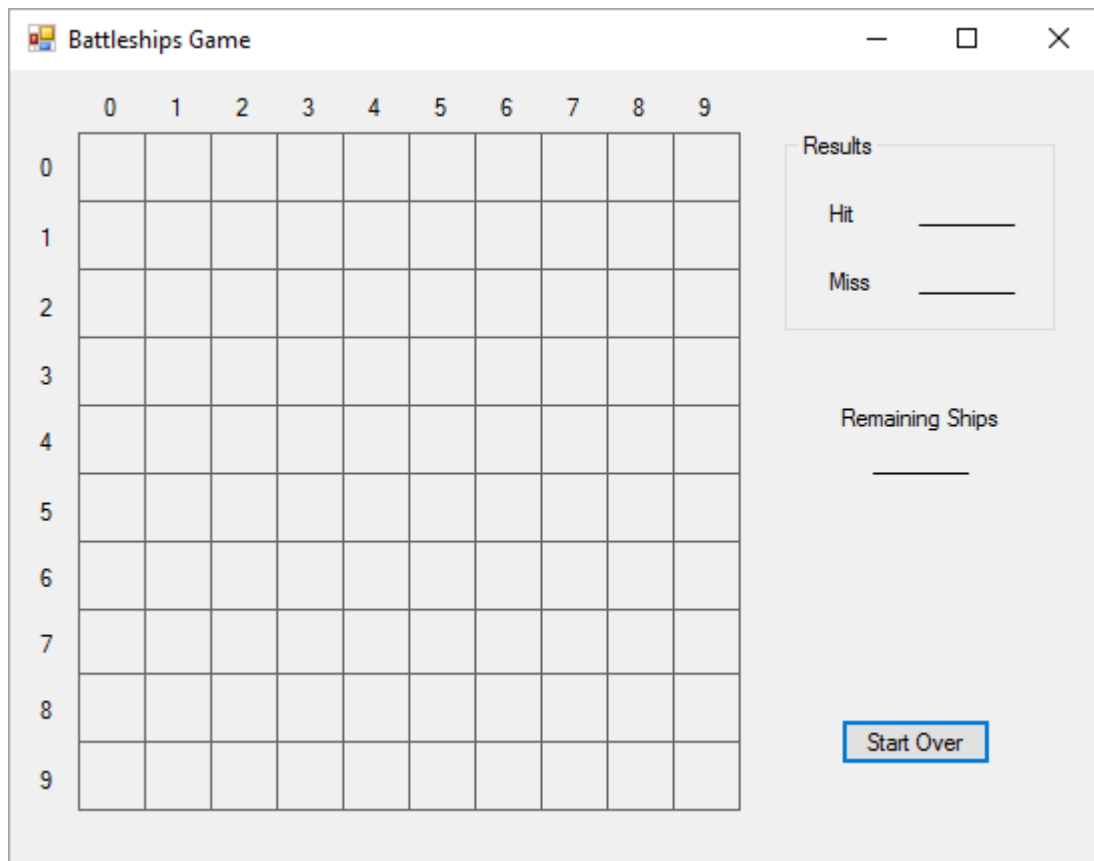
Late Submission Penalty

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

Assessment Criteria

Option 1

Develop a “Battleships” game. Use a 2D array to make a grid, an at least 10x10, to play battleships on. Create a suitable interface so users can guess an X and Y coordinate.



The requirements are:

1. Ships can be in any grid size, either horizontally, vertically or diagonally.
2. When the game starts, place a random number of ships in a randomised grid coordinate, all invisible to the user. The ships may be of different graphic design with varying grid size, e.g. submarine, patrol boat, cruiser, battleships, etc.
3. The user will select a coordinate to locate the ships.
4. Give indication to the user whether the selected coordinate is a “miss” or a “hit”, and record the score on a scoreboard.
5. Start Over button resets the game.

Option 2

Develop a software application in accordance to the requirements below:

The screenshot shows a 'Sales System' application window with the following sections:

- Customer Details:** Fields for First Name, Last Name, Street Address, Suburb, City, and Country.
- Credit Card Information:** Fields for Name on Card, Card Type (dropdown), Card Number, Expiry (month/year dropdowns), and Issuing Bank (dropdown).
- Product Selection:** A table with columns for Product Name, Unit Price, and Quantity. Products listed are Shoes, Jackets, Gloves, Beanies, Sweaters, and Scarfs, each with a checkbox.
- TOTAL AMOUNT:** A label followed by a text input field for the total amount.
- Payment Type:** Radio buttons for Cash, Credit Card, and Bank Transfer.
- Buttons:** 'Reset' and 'Show Summary'.

The requirements are:

1. For Customer Details, First Name and Last Name are mandatory.
2. Credit Card Information section should be disabled, hence, preventing use of objects within the section, when Payment Type selected is not Credit Card.
3. If Credit Card is selected as Payment Type, all information on Credit Card Information are mandatory.
4. When the application loads, populate the following objects:
 - a. Card Type with Visa, MasterCard, Diners Club, American Express
 - b. Expiry with months (January to December), and years (current year to next 10 years, dynamically)
 - c. Issuing Bank with New Zealand and Overseas
5. Product Selection should allow selection of Product Name. It is mandatory to have a minimum of one Product Name selected and one Quantity entered. Unit Price is to be added thru program code when application loads.
6. TOTAL AMOUNT is calculated based on the Product Name selected, Unit Price, and Quantity entered.
7. If the TOTAL AMOUNT calculated is below \$50, Credit Card option is disabled. Cash is disabled when TOTAL AMOUNT exceeds \$1,500.
8. Show Summary button displays all the information on a message box, if all the requirements are fulfilled; otherwise, displays an error message using a message box.
9. Reset button set the application to start-up state.

The features to include are:

1. Proper naming convention used
2. Meaningful internal documentation
3. Meet the interface/template requirement
4. Use of functions and procedure
5. Correct calculation logics
6. Use of decision statements, i.e. IF or SWITCH
7. Input verification and error checking
8. Suitable variables/arrays 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.

-- THE END --