Learning Outcomes:
On conclusion students should be able to:

- Develop a problem-based strategy for creating and applying programmed solutions
- Create, edit, compile, run, debug and test programs using an appropriate development environment

FERRY TICKETING SYSTEM

A small ferry company has just purchased a computer for its new automated ticketing system. The company director has asked you to design the new system to assign seats for each trip of the 50-seater ferry, which covers the route from Penang to Langkawi and back daily. The upper deck of the ferry is reserved for business class passengers and can accommodate 10 people. The main deck of the ferry, the economy class can accommodate 40 people. Assume the company has at least 8 ferries - each with a different Ferry ID, which travel at one-hour intervals from 10 am to 5 pm daily.

SECTION A : BASIC REQUIREMENTS OF THE SYSTEM

1. **Main Menu**
Your initial program design should display the following menu alternatives:

FERRY TICKETING SYSTEM
P – to Purchase Ticket
V –to View Seating Arrangement
Q – to Quit the system

2. **Submenu**
The following submenu will be displayed when P is selected:

PURCHASING MODULE
B – to purchase ticket for Business class
E – to purchase ticket for Economy class
M – to return to Main Menu

3. **Assigning Seats**
If the person types B, then your program should assign a seat in the business class (seats 1-10). If the person types E, then your program should assign a seat in the economy class (seats 11 - 50).
The following submenu will be displayed when V is selected:
SEATING ARRANGEMENT MODULE
F- to select Ferry ID
T- to select Trip Time

4. **Boarding Ticket**
   Your program should then print a boarding ticket indicating the person’s name, seat number, whether it is in the business or economy class of the ferry, Date and time of departure, Source and Destination of the trip and Ferry ID.

5. **Seating Chart**
   Use a list to represent the seating chart of the ferry, indicating the availability of the seats within each trip of the ferry. Initialize all the elements of the list to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding elements of the list to 1 to indicate that the seat is no longer available. Your program should never assign a seat that has already been assigned.

   The **Ferry ID** will be requested when V is selected from the main menu and the seating arrangement for that ferry will be displayed in a tabular form, e.g.

```
***********************************************************
*     Ferry ID : 007                 Date: 19 Sept 2005   *
***********************************************************
*     BUSINESS CLASS                                      *
***********************************************************
*     1     *     1     *     1     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     ECONOMY CLASS                                       *
***********************************************************
*     1     *     1     *     1     *     1     *     1     *
***********************************************************
*     1     *     1     *     1     *     1     *     1     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
*     0     *     0     *     0     *     0     *     0     *
***********************************************************
```

6. **Alternative seating**
When the business class is full, your program should ask the person if it is acceptable to be placed in the economy class (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message “Next trip leaves in 1 hour”.

INSTRUCTIONS

This is a group assignment. Each group should consist of 2 – 3 members only. Upon submission of your assignment, you would be required to present your assignment at a date and time specified by your module lecturer.

Each team member is required to contribute towards all sections of the assignment, present and explain his or her contribution of the work done. Each team member should also be able to answer questions posed with regards to the project and / or subject matter.

DELIVERABLES

You are required to submit:

a. A softcopy of the program coded in Python – submitted on a CD. The program should include the following:
   - Basic programming concepts such as displaying and reading of inputs, displaying of outputs, declaration of variables and assignment of values, comments – to explain various parts of the program, etc.
   - Control structures i.e. selection and iteration
   - Lists and functions
   - File I/O

b. Documentation of the system, that incorporates basic documentation standards such as header and footer, page numbering and which includes
   - Cover page
   - Table of contents
   - Workload matrix to indicate the contribution of each individual for each required component (shown in %age form) and signed off by each team member
   - Design of the program – **using pseudocode or flowcharts** – which adheres to the basic requirements listed above
   - Source code of the program in Python not to be included in documentation.
   - Sample outputs when the program is executed with some explanation of the outputs / sections of the program
   - Additional features which have been incorporated in the solution in terms of design and Python
   - Assumptions
   - References

The documentation should be comb-bound with the CD attached.

ASSESSMENT CRITERIA

Individual Component (50%)

- Algorithm (10%)
- Presentation & QA (15%)
- Understanding of programming constructs (15%)
- Contribution (10%)

Group Component (50%)

- Design (20%)
- Module Integration (10%)
- Documentation (20%)