

***C# BASICS***

**Training Assignments**

| **Document Code** | **25e-BM/HR/HDCV/FSOFT** |
| --- | --- |
| **Version** | **1.1** |
| **Effective Date** | **20/11/2012** |

**Hanoi, 06/2019**

**RECORD OF CHANGES**

| **No** | **Effective Date** | **Change Description** | **Reason** | **Reviewer** | **Approver** |
| --- | --- | --- | --- | --- | --- |
|  | 01/Oct/2018 | Create new | Draft |  |  |
|  | 01/Jun/2019 | Update template | Fsoft template | DieuNT1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Contents**

[Day 4: Assignment 13: C# Basic Project Practice 4](#_heading=h.1fob9te)

[Objectives: 4](#_heading=h.3znysh7)

[Business needs: 4](#_heading=h.2et92p0)

[Prerequisites: 4](#_heading=h.tyjcwt)

[Technologies: 4](#_heading=h.3dy6vkm)

[Stored Data: 4](#_heading=h.4d34og8)

[Functional Requirement: 5](#_heading=h.2s8eyo1)

[User Interface Requirement: 6](#_heading=h.17dp8vu)

|  | **CODE: Net.M.A013**  **TYPE: MEDIUM**  **LOC: 190**  **DURATION: 90 MINUTES** |
| --- | --- |

# Day 4: Assignment 13: C# Basic Project Practice

**Objectives:**

* Understand and practice with Advanced OOP in C#.
* Understand and apply with Collection classes in C#: List, Dictionary, Hashtable.
* Understand and practice with Exception handing, I/O.

**Business needs:**

* Create a Window Console application bases on, OOP, Exception Handing, IO, Collection to manage an Airport (Airport Management System) with **fixed wing** **airplane** and **helicopter**

**Prerequisites:**

* Working environment: Visual Studio 2013 or higher.Practice code in Visual Studio
* **Delivery**: Source code packaged in a compress archive.

**Technologies:**

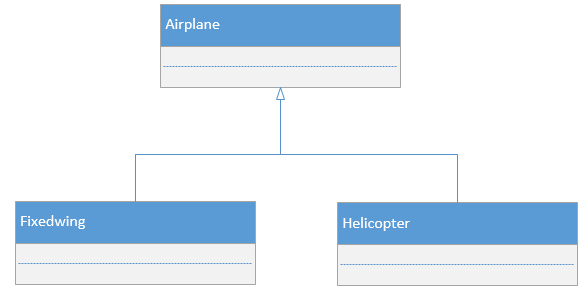
The product implements one or more technology:

* C# Basic
* Collection
* OOP
* Exception Handing

**Stored Data:**

* All data information will be saved to the file.

For the class hierarchy is as follows, the trainee let's create the classes install this class diagram to be able to relationship between it.



* The Fixedwings class contains the information about fixed wing airplanes. Each fixed wing airplane has its ID, Model, Plane Type, Cruise Speed, Empty Weight, Max Takeoff Weight, Min Needed Runway Size and Fly method (“fixed wing”)
* The **Helicopters** class contains the information about helicopters. Each helicopter has its, ID Model, Cruise Speed, Empty Weight, Max Takeoff Weight, Range and Fly method (“rotated wing”)
* The airports class contains the information about 4 airports. Each airport has an ID, Name, Runway Size, Max Fixedwing Parking Place, List of Fixedwing Airplane ID, Max Rotatedwing Parking Place, List of Helicopter ID.

Program requirements must validate the properties:

* ID is a string of 7 characters, started by “FW” for fixed wing airplane, “RW” for helicopter and “AP” for airport, followed by 5 digits. ID is unique.
* The model size is maximum 40 characters.
* Three fixed wing airplane type are CAG (Cargo), LGR (Long range) and PRV (Private).
* If a fixed wing airplane is parked in an airport, its min runway size does not excess the airport runway size.
* The max takeoff weight of helicopter does not excess 1.5 times of its empty weight.

**Functional Requirement:**

* The program has the function to create and to delete an airport. The name, runway size and capacity of the new airport are selected when a new airport is creating.
* The program has the function to add to an airport one or more fixed wing airplane(s) which currently does not participate to an airport and which has the min needed runway size shorter than the airport runway size.1
* The program has the function to remove one or more helicopter(s) from an airport.
* The program has the function to add to an airport one or more helicopter(s) which currently does not participate to an airport.
* The program has the function to remove one or more helicopter(s) from an airport having.
* The program has the function to change plane type and min needed runway size of fixed wing airplane.

**User Interface Requirement:**

The program has a screen console for UI

The main screen allows selecting the functions for:

* Input data from keyborad
* Airport management
  + Display list of all airport information, sorted by airport IDA
  + Display the status of one airport, selected by airport ID
* Fixed wing airplane management
  + Display list of all fixed wing airplanes with its parking airport ID and name
* Helicopter management group
  + Display list of all helicopters with its parking airport ID and name
* Close program

Each unsuccessful action will be informed to user by an error message.