

## Assignment One

Trimester 2, 2024

### Develop and Test Benchmark Project

Weighting: 30%

**Due: Friday, 9<sup>th</sup> August**

#### Assessment

Submit a soft copy of your Benchmark Project in C# and test documents through digital Dropbox in Moodle. **All code must be commented**, this includes comments for every class, method, constructor and inline comments. Your test documents must include all the test cases and your judgment and comments when using the following required testing methods.

#### Purpose of this Assignment

The intention of the benchmark application is to apply the majority of software testing skills to a project that you have developed based on a sample program in this course. You must design your own test documentation for the following four testing methods.

- 1) Black-box testing: You must report your **test cases** for all the functional requirements of your project, e.g., buttons, menus, mouse move over, etc.
- 2) White-box testing: You must report your test cases for all the non-functional requirements of your project such as security, reliability (e.g., validations, boundary data, etc.).
- 3) Grey-box testing: You must report your test cases for all the object-oriented classes (**constructors and methods**) using Unit Test.
- 4) Automated Testing: You must explain and demonstrate how you use Automated Testing.

#### Project Description

Based on the sample program of this course, you are required to develop a Benchmark Application for managing personal information of staff members in a business. The information is not limited to staff Id, name, date of birth, email, annual salary, etc.

The project consists of four (4) classes: Form1.cs, FileManager.cs, Filter.cs, and MyClass.cs.

**Form1.cs** is the GUI for the application. You will make your own GUI design.

**MyClass.cs** is the class used to store staff data about objects (used throughout the application).

**FileManager.cs** is the class used by the application to **read and write** objects to and from a simple text file (.txt).

**Filter.cs** is the class used by the application to search and sort lists of objects (Search and Sort buttons). The searching and sorting methods must use LINQ lambda or query expressions.

### Program Code and Comments

The Benchmark Application must provide the following functions:

1. Only validated personal object data can be input and displayed on a List Box.
2. The selected object data on List Box can be **deleted**.
3. The objects data on the List Box can be appended to the end of existed data in a **text file**.
4. All the objects from text file can be loaded to the List Box.
5. Sort all the objects data ascending and deciding.
6. Search for objects based on a keyword.

All code must be **commented**, for **every class** and class **member** and inline comments.

### Unit Testing

A Unit Testing project must also be added to the Benchmark Application solution. The Unit Testing solution must reasonably test every method of the Filter class, every member of the MyClass, and every method of the FileManager class. Multiple test cases can be used for testing a single method. Unit test methods should also be commented.

### Marking Guide

Section	Weighting
GUI + Source File	10%
Staff Class (Code and Comments)	10%
File Manager Class (Code and Comments)	10%
Filter Class (Code and Comments)	10%
Black-box testing	15%
White-box testing	15%
Grey-box/Unit Testing	20%
Automated Testing	10%

\* Marks for Code and Comments will generally weighted evenly, e.g. 50% for correct code, and 50% for correct comments. Sections of missing code will also lose marks for the corresponding missing code.

\*\* Marking for testing will generally weighted on test cases (50%), and judgment and comments (50%).

\*\*\* Weighting percentages are indicative only and may vary on the final marking schedule

### Plagiarism

Copying the work of others, or using other people's ideas as your own without acknowledging the source is called plagiarism. Lecturers will not accept such work and you may be penalised by losing marks or failing an assessment.

All individual assignments and tests must be entirely your work. Discussion and assistance between students who are working in groups is ok, but all work handed in must be your work and written in your own words, except for assignments based on group achievement. To reinforce this, you are required to sign the declaration on the cover sheet of each assignment. Further information is in the Guidelines for Written Assignments handbook and the Faculty of Business and Information Technology Student Handbook.