MODULE NAME:	MODULE CODE:
PROGRAMMING 2B	PROG6212

ASSESSMENT TYPE: ASSIGNMENT 2 (PAPER)

TOTAL MARK ALLOCATION: 100 MARKS

TIME ALLOWANCE: 15 HOURS

TOTAL PAGES: 5

TOPIC: LEARNING UNIT 2: WORKING WITH FILES

STUDENT NAME:

STUDENT NUMBER:

INSTRUCTIONS:

1. ***No more than 25% of the assignment may be copied from original source(s) used, even if referenced correctly.***
2. *This is a practical assignment.*
3. *Assignments must be created using Microsoft Visual Studio 2010 and saved as appropriately named C# files.*
4. *Insert comments in your code which will identify you by name and student number.*
5. *All work must be adequately and correctly referenced. Code segments copied directly from a source must be referenced.*
6. ***Ensure that you follow good programming coding standards.***
7. ***This is a non-specific assignment; therefore, marks will be awarded for creativity and effort given.***
8. *This is an individual assignment. All work must be your own work.*
9. ***Make a copy/backup of your assignment before handing it in.***
10. *Follow all instructions on the assignment cover sheet.*

© The Independent Institute of Education (Pty) Ltd 2013

Assignment two has been set to demonstrate your C# programming abilities. You are required to use your knowledge of at least chapters one to thirteen in the prescribed text book with the emphasis on chapters ten to thirteen. In other words, ensure that you demonstrate your understanding of: event handling; OOP features, exception handling and files.

Please note that when completing your assignment, you are required to use internationally accepted coding standards (see page 112 in the prescribed textbook). Include comprehensive comments explaining variable names, methods and the logic of your programming code. Document your application thoroughly. Although you are not required to submit user and developer type documentation with this assignment you are required to submit it with your Portfolio of Evidence and so it would make good sense to create this documentation while you are in the process of developing the application.

The types of system documentation you should at least consider are:

- Description of the system (Describe the problem and how it can be solved using a file-based system (Maximum 5 pages) to be submitted with your PoE);
- System and program flowcharts;
- UML diagrams (at least class and use-case diagrams);
- Test data used to debug the application;
- Help files/ user guides for the system.

Application Specifications (Marks: 100)

Choose any problem that can be solved using a file-based application:

- Design the files required for your system. You will need a file used to store data and at least two (2) files to contain management reports. The minimum requirement is one (1) storage file and two (2) management report files but you may use as many as is necessary for your application;
- The storage file/s must be random access file/s;
- Create the file/s using programming code. (Ensure that when you submit your assignment that you include copies of these created storage and report files and they can be located by your lecturer. Provide the access path where the files are located as part of your documentation);

© The Independent Institute of Education (Pty) Ltd 2013

- The storage file must consist of records. In other words, instantiate objects of a particular class type. The object represents a record. The data in this file will be used in your application. E.g the class type can be Patients, the objects instantiated can be Maternity, surgical, paediatric. This means that each record in a particular file will be related to either maternity patients, surgical patients or paediatric patients.

In conclusion:

Your application must do the following:

- Create at least one (1) randomly accessed file using program code (to store data). The file must be able to have new data appended to existing data. i.e. additional records can be added at a later stage. Input must be done using a GUI form with textboxes and buttons to write to the file;
- Contain GUI pages that allow the user to view selected data (a record) stored in the file on a form (not another file);
- Give the user a minimum of two (2) report options;
- Use exception handling to determine if the files or any input is valid. Make use of MessageBoxes to display error messages;
- Use menus to navigate between forms.

Ensure that you have covered all items listed in the marking rubric below. The mark assigned will follow the following sliding scale:

0 Mark	Not completed/ not submitted;
1 Mark	Attempted but incorrect;
2 Marks	Attempted, logic/ process in the correct direction;
3 Marks	Logic/ process correct;
4 Marks	Logic/ process and coding correct;
5 Marks-	Excellent, Logic/ process correct and coding correct and efficient. Student has gone above and beyond specified requirements.

© The Independent Institute of Education (Pty) Ltd 2013

Marking Guidelines

Criteria	Possible mark	Mark awarded
Good coding standards		
• Comments / code readability throughout C# code;	5	
• C# file naming/modular coding – using different C# files for different modules; and	5	
• Efficient code (no redundancy).	5	
Subtotal	15	
File (storage)		
• Creation of at least one random file;	5	
• File stores records;	5	
• File can be read from, and appended to.	5	
Subtotal	15	
File (report/s)		
• Creation of at least two report files;	5	
• File displays report in correct format (columns aligned with headings).	5	
Subtotal	10	
GUI Input		
• Data entered using GUI objects;	5	
• Button click writes to file;	5	
• Validation of input data.	5	
Subtotal	15	
GUI Output of file contents		
• User can select a record to display;	5	
• Button click displays record.	5	
Subtotal	10	
Application page(s)		
• Management reports produce correct information;	5	
• Customised class/ static methods;	5	
• Menus;	5	

© The Independent Institute of Education (Pty) Ltd 2013

• Error messages in MessageBoxes.	5	
Subtotal	20	
Other Marks • Student's application is user-friendly and incorporates good aesthetic design; • Creative application; • Student has used advanced features not covered in class	 5 5 5	
Subtotal	15	
Total	100	

[TOTAL MARKS: 100]

© The Independent Institute of Education (Pty) Ltd 2013