

Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

GX1523

Semester II Examinations, 2003/2004

Exam Code(s)	1SD1 1MF1 4BP1
Exam(s)	HIGHER DIPLOMA in APPLIED SCIENCE (SOFTWARE DESIGN & DEVELOPMENT) M.Sc. in SOFTWARE DESIGN & DEVELOPMENT B.E. in ELECTRONIC & COMPUTER ENGINEERING
Module Code(s)	CT470 CT520 CT862
Module(s)	GUI AND OBJECT ORIENTED PROGRAMMING OBJECT ORIENTED PROGRAMMING
Paper No.	1
Repeat Paper	Special Paper
External Examiner(s)	Prof. D. Bell
Internal Examiner(s)	Prof. G. Lyons Dr. S. Redfern Dr. M. Schukat

Instructions:

Instructions for candidates presenting for

CT862 and CT520:

Time allowed: 3 hours

Answer any 2 questions from section A

and

Answer any 2 questions from section B

All questions carry equal marks.

Please use separate answer books for each section.

Instructions for candidates presenting for CT470 only:

Time allowed: 2 hours

Answer all 3 questions from section A

Duration	2 hrs or 3 hrs
No. of Answer books	2

Requirements:

Handout

MCQ

Statistical Tables

Graph Paper

Log Graph Paper

Other Material

No. of Pages

9

Department(s)

Information Technology

Section A (Object Oriented Programming)

Programs are to be written in C++

Q.1.

(i) 8 marks

Correct the code fragment below, which contains 7 syntax errors and one possible runtime error. You'll get 1 mark for every correct answer; for every incorrect answer 1 mark is subtracted. The total amount of marks you can achieve is between 0 and 8, i.e. you can't get a negative overall mark.

```
%include <iostream.h>

class parent {

private:
    int size;
    int* array;

public:
    // Constructor
    void parent(int k) : size(k) {
        array = new int(k);
    }

    void ChangeArray(int index, int val) {
        array[index] = val;
    }

    // Destructor
    ~parent(void)
    {
        delete size;
    }
}

class child : private parent {

private:
    int index;

public:
    // Constructor
    child(int a, int b, int c) : parent(a, b) {
        index = c;
    }

    int get_size(void) {
        return size;
    }
};
```

(ii) 4 marks

What are the benefits of *exception handling* in C++? Provide a code fragment to illustrate your answer.

(iii) 8 marks

Prototype and implement a **fraction** class which has as attributes two integer variables called *numerator* and *denominator*.

Overload the plus ("+"), the division ("/") and the stream insertion ("<<") operator.

Hints:

$$a / b + c / d = ((a * d) + (b * c)) / (b * d)$$

$$(a / b) / (c / d) = (a * d) / (b * c)$$

The stream insertion operator is based on the following function prototype:

```
friend ostream& operator<<(ostream& os, const fraction& f);
```

Q.2.

(i) 8 x 2.5 marks

With reference to the C++ language, explain the meaning/purpose of the following terms:

- (a) static_cast
- (b) static const attributes
- (c) function overloading
- (d) default arguments
- (e) ->
- (f) namespace
- (g) this
- (h) >>

Q.3.

(i) 6 marks

Distinguish between *compile-time polymorphism* and *run-time polymorphism*. Use examples to illustrate your answer.

(ii) 8 marks

What are *constructors* and *destructors* useful for?

Explain how a constructor of a base class can be called *explicitly* or *implicitly*, when an instance of a derived class is created.

Implement the constructor and the destructor for the following class:

```
class car
{
private:
    char *owner; // Pointer to char array, which contains name of owner
    char *make;  // Pointer to char array, which contains make of car
    int model;   // Model of car.

public:
    // What to implement in constructor:
    // A) Allocate char arrays on the heap; use strlen (see hints below)
    //    to determine the size of theOwner and theMake.
    // B) Copy char arrays using strcpy (see hints below).
    // C) Update attribute model.
    car(char* theOwner, char* theMake, int theModel);

    // Destructor
    ~car(void);
};
```

Hints:

*int strlen(char *p)* returns the length of the string passed without the terminating character “\0”.

You probably need the string copy function *strcpy(char *destination, char *source)*.

(iii) 6 marks

What is the meaning of the term “class inheritance”? Explain the difference between deriving a class using the keyword **public** and deriving a class using the keyword **private**.

Section B (GUI Programming)

Programs are to be written in Visual Basic (VB) unless otherwise stated

Q.4.

Write a program that provides a multi-line text box into which the user can enter text. The program should then calculate the *Flesch Readability Index*, as described below, in order to estimate the legibility of the text that has been entered.

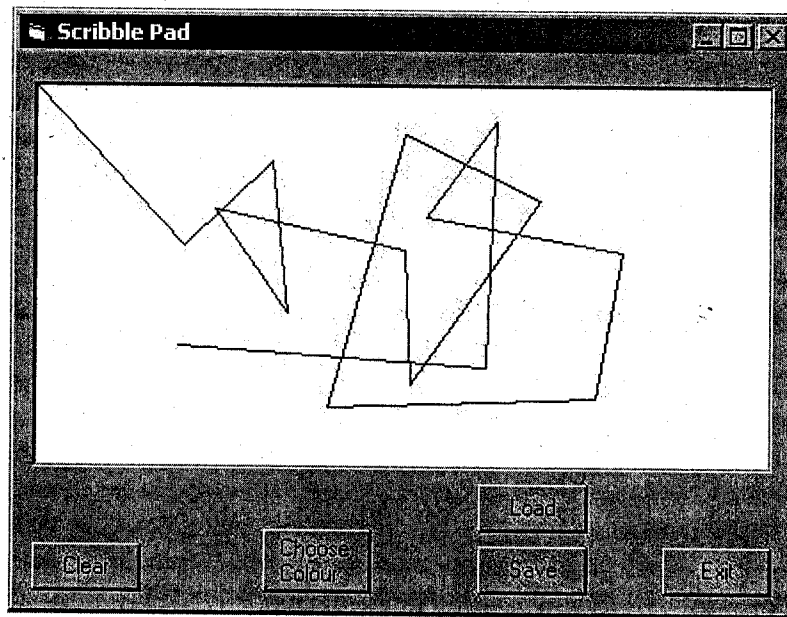
The *Flesch Readability Index* was invented as a simple tool for determining the legibility of a document without linguistic analysis. It may be implemented using the following 4 steps:

1. Count all words. A *word* is any sequence of characters delimited by white space.
2. Count all syllables in each word. Each *group* of adjacent vowels (a, e, i, o, u, y) counts as one syllable (for example, the "ea" in "real" contributes one syllable, but the "e..a" in "regal" counts as two syllables). However, an "e" at the end of a word doesn't count as a syllable. Also, each word has at least one syllable, even if the previous rules give a count of 0.
3. Count all sentences. A sentence is ended by a full stop, colon, semicolon, question mark, or exclamation mark.
4. The index is computed by the following formula:

$$Index = 206.835 - 84.6 * \frac{\text{syllables}}{\text{words}} - 1.015 * \frac{\text{words}}{\text{sentences}}$$

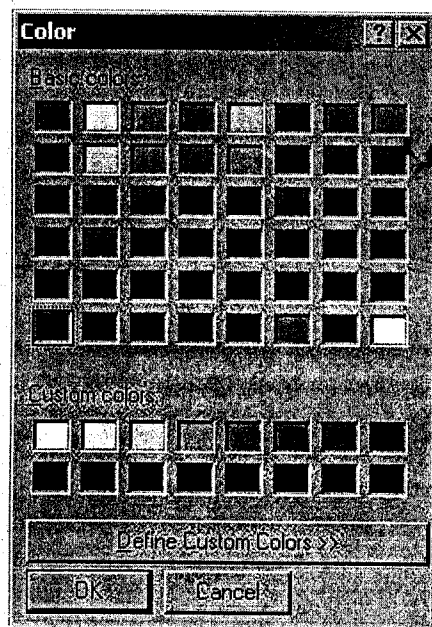
Q.5.

Write a “Scribble Pad” application as illustrated below. A PictureBox control displays lines drawn by the user by clicking and moving the mouse on it. Each mouse click should result in a line being drawn from the position at which the mouse was previously clicked, to the position at which the mouse is now located.



The “Clear” button should remove all lines from the PictureBox.

When the user clicks the “Choose Colour” button, the program should display a colour-selection dialog box (as shown below), and the selected colour should be used as the colour of the lines drawn after that.



The program should also allow the user to load and save their drawing to/from disk using a sequential access text file. The user, via the use of a standard Windows load/save dialog box, should supply the name of the file.

The Exit button should end the program.

Q.6.

Write a program that uses an MS FlexGrid control to display and edit stock control information for a group of garages.

(15 marks)

- Each stock type occupies a row of this grid. The name of the item is shown in column 0.
- Columns 1 through 4 (i.e. Warehouse, Shop1, Shop 2, Shop 3) show the levels of stock held at each of four different geographical locations.
- Column 5 shows the value of a single unit of stock.
- Ensure that the user can only type numbers into this grid, and only into columns 1-5. Columns 0 and 6 should ignore key-presses.
- Column 6 displays the total value of stock, and is calculated as the sum of columns 1-4, multiplied by the value in column 5.
- When the user clicks "add new row", a new row is added to the bottom of the grid. The user is asked for the name of the product to put into column 0 (the grey cell) of the new row.

(5 marks)

When the user clicks "Open in MS Word", the program should:

- Create and open an instance of Microsoft Word,
- Create a blank Word document
- Add a new table of appropriate dimensions to that document
- Write the contents of the grid into that table.

	Warehouse	Shop1	Shop2	Shop3	Unit Value	Total Value
Apple	20	9	3	4	200	7200
Banana	45	12	5	2	20	1280
Orange	30	10	5	4	50	2450
Watermelon	12	8	4	5	50	1450
Pineapple	5	3	1	2	1000	11000

Hints:

Assuming that oWord is an object variable that references a valid instance of MS Word, you can use the following VB code to create a new table with 2 rows and 5 columns:

```
oWord.ActiveDocument.Tables.Add NumRows:=2, NumColumns:= 5
```

You can use the following code to move the cursor to the right by one table cell. (Note: moving right from the right-most column will move the cursor into the next row on the left-most column):

```
oWord.Selection.MoveRight Unit:=wdCharacter, Count:=1
```


CT862 VB Information Page - attach to exam paper

Object	Important Properties	Important Events & Methods
Check Box	Value	Click
Clipboard		GetText; SetText; Clear
Combo Box	Sorted; Text; List(index); ListCount; SelStart; SelLength; SelText	AddItem(item, index); RemoveItem(index); Clear
Command Button	Caption	Click
Common Dialog	FileName; Filter; Color; Flags; FontBold; FontItalic; FontStrikeThru; FontUnderline; FontName; FontSize; Copies; FromPage; ToPage	ShowOpen; ShowSave; ShowColor; ShowFont; ShowPrinter
Form	Caption; BackColor; AutoRedraw	Show; Hide; Load
Frame	Caption	
FlexGrid	Rows; Cols; FixedRows; FixedCols; Row; Col; Text; ColWidth(index); TextMatrix(row,col)	KeyPress(KeyAscii As Integer)
Image, PictureBox	Picture	
Label	Caption; AutoSize; WordWrap	
List Box	Sorted; List(index); ListCount; Columns; MultiSelect; Selected(index)	AddItem(item, index); RemoveItem(index); Clear
Menu item	Caption; Enabled; Checked	Click
OLE Container	Class; SourceDoc; SourceItem	Update; SaveToFile; ReadFromFile
Option Button	Value	Click
Printer		Print; EndDoc
Scroll Bar	Value; Max; Min; LargeChange; SmallChange	Change; Scroll
Text Box	Text; MultiLine; ScrollBars; BackColor; ForeColor; FontBold; FontItalic; FontStrikeThru; FontUnderline; FontName; FontSize; SelStart; SelLength; SelText	KeyPress(KeyAscii As Integer); Change
Timer	Interval; Enabled	Timer

Miscellaneous Object Events:

GotFocus; LostFocus; Enabled; Visible; TabIndex; MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single); MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single); MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Miscellaneous Object Methods:

SetFocus; Move(x, y); Line (x1,y1) - (x2,y2)

Miscellaneous Object Properties:

Width; Height; Left; Top