

Computing (9691)

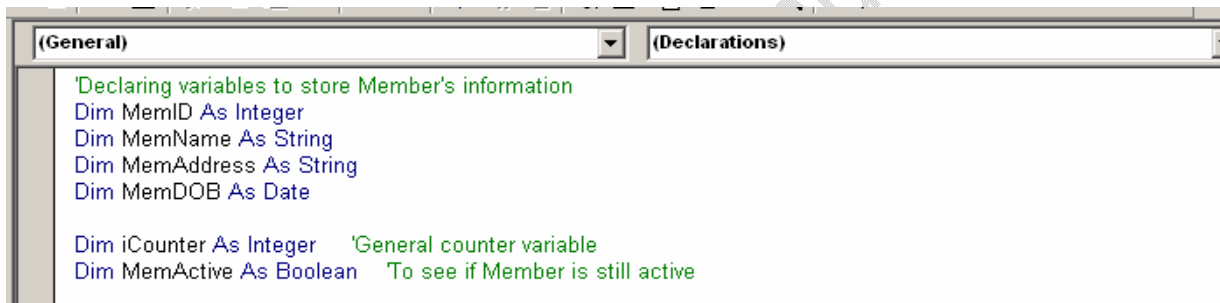
Module 2: Practical Programming Project Guidelines

Guidelines for the Project Software

1. The developed software must demonstrate the following eight programming techniques along with comments to highlight their usage in the program code.
 - a. Multiple data types
 - b. Selection statement
 - c. Iteration (Loop) statement
 - d. User defined Procedure
 - e. User defined Function
 - f. Searching
 - g. Files
 - h. Arrays/Records
2. Program code must be well annotated. Use brief, meaningful comments to highlight important sections/processes in the code. Example:

Open StudentFile.txt For Input As #1 'Opening student file in Input mode to read data from it

3. Short and meaningful names must be used for variables, controls, objects, functions, procedures and user defined data types. Labels such as i, p, x etc. will not get any marks. Example:



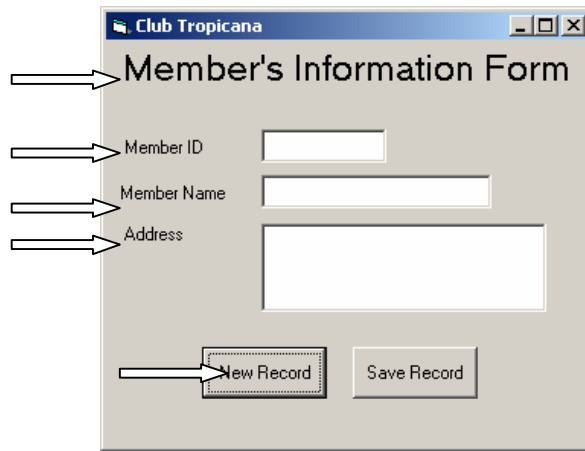
The screenshot shows the 'Declarations' window in a Visual Basic IDE. It contains the following code:

```
'Declaring variables to store Member's information  
Dim MemID As Integer  
Dim MemName As String  
Dim MemAddress As String  
Dim MemDOB As Date  
  
Dim iCounter As Integer 'General counter variable  
Dim MemActive As Boolean 'To see if Member is still active
```

4. Each Form's code must have a header with a description of the form. Example:

```
*****  
'Form Name: frmMembers  
'Purpose: Allows the user to enter, edit and view information regarding club members.  
'Author: Ali Saeed  
'Language: Microsoft Visual Basic 6  
'Center: Central School System (PK-980)  
*****
```

5. Make sure all controls on the form have proper labels to show what these controls are for. Example:



6. Controls, especially Text Boxes, command Buttons and Forms, should be of appropriate size. Controls must not be too big or too small for the data expected. There should be a consistency in the height of controls, especially the Text Boxes and Command Buttons. The size and type of Forms must be carefully chosen making sure that there is not unnecessary empty space in the Form.
7. All Text Boxes must have their MaxLength property set according to the data that is to be entered in them. As a general guideline you can use these ranges:

All Name Text Boxes = 25 to 35 characters

All Address Text Boxes = 40 to 50 characters. Turn MultiLine property on.

All Money Text Boxes = 4 to 6 characters (more is OK if your particular project requires it)

All Email Text Boxes = 30 characters

All Phone Text Boxes = 7 (12 if you have to include city code as well, but it's not necessary in most cases)

All Cell Phone Text Boxes = 11 characters

All Date fields = Use DTPicker control with MinDate and MaxDate set accordingly

All ID Text Boxes (e.g. Student ID) = 2 to 3 characters

8. Input Controls should have, where appropriate, data validation routines. As general guidelines use the following:

Fields/Controls	Example	Validation Type
All Name Input Fields	Student Name, Client Name	Alphabets, space and backspace only
All Address Input Fields	Company Address	No validation
All Phone Input Fields	Home Phone, Fax, Cell Number	Numeric and backspace only, no space
All Date Input Fields	Date of Birth, Shipment Date	Use DTPicker as dates are validated automatically
All Email Text Fields	Client Email	No validation
All Money Fields	Salary, Invoice Total, School Fee	Numeric and backspace only, no space
All Gender Fields	Male, Female	Use Option Buttons
All ID Text Fields	Student ID, Employee ID	Numeric only, no space

9. Menus, if used, must have clear and logical item names. If Command Buttons are used as a menu, then their labels must be clear and descriptive.
10. The coding must employ suitable indentation to clearly distinguish the programming constructs used. Example:

```
(General) (Declarati)
'Declaring variables to store Member's information
Dim MemID As Integer
Dim MemName As String
Dim MemAddress As String
Dim MemDOB As Date

Dim iCounter As Integer 'General counter variable
Dim MemActive As Boolean 'To see if Member is still active

Private Sub cmdNewRec_Click()
If txtName.Text <> "" Then
MemID = Val(txtID.Text)
MemName = txtName.Text
MemAddress = txtAddress.Text
If txtDOB > #12/21/2005# Then
MemDOB = txtDOB.Text
Else
MsgBox "You are too young to apply.", vbOKOnly, "Error"
End If
End If
End Sub
```

11. When copying code into the documentation, all portions of the code including procedures and functions must be clearly names and well separated with full annotation.
12. Software should not include any libraries and/or third-party controls that are not easily transferable to the client's machine. In case any pre-built or third-party controls/code libraries are used, it must be clearly mentioned in the code (in shape of annotations) and also in the documentation.
13. The software must be a working application with a practical purpose. Students must be able to demonstrate that their software can produce meaningful results which can be reproduced by a person testing their software.
14. It is recommended that some kind of online help is included with the software. It can be in shape of a screen showing instruction regarding basic operations of the software.

Mubashir N. Mir - www.mubashir.com

Guidelines for the Project Documentation

1. The documentation must be word processed with consistent formatting used throughout the documentation.
2. Choose A4 as your document size.
3. Use only non-comic and non-script fonts such as Arial, Times New Roman or Garamond. Do not use more than two fonts in your documentation (one for headings and one for text).
4. Recommended font sizes are:
 - a. Section Headings: 25 – 30 points in bold.
 - b. Main Headings: 18 – 20 points in bold.
 - c. Sub Headings: 15 – 18 points in bold.
 - d. Document Text: 11 or 12 points.

Example:

Section 1: Analysis

Requirements Definition

a) User Requirements

The user requires an easy to learn software that fulfills the following requirements.

5. The document margins (top, bottom, left and right) should not be more than 1 inch.
6. Documentation must be in single line spacing.
7. Each section and sub section must be separated visibly by proper headings.
8. Documentation text must be either Left Justified or Justified. Only headings and Tables can be Center Justified.
9. Avoid one line paragraphs and very long sentences.
10. Make a proper title page at the beginning of the documentation.
11. Make a Table of Contents with page numbers for reference.

In order to gain high marks, the following project requirements must be met.

Problem/Task Identification

[2 marks]

1. A full description of the problem to be solved along with the data to be input and the desired output. (Note: data input and output description can be described in separate sections.) You will need to describe:
 - a. what the problem is
 - b. how and where it originates
 - c. who gets affected by it and how
2. Input Requirements (this should include names of data fields and their purpose).
3. Output Requirements (this should include names of reports along with the fields that are to be shown).
4. User Requirements (this should include any functional requirements from the user such as "the software should be user friendly", "the software should enable the user to take backups of the data", "dates should be shown in dd/mm/yyyy format", etc.)

Program Design

[6 marks]

1. A logically correct, detailed and complete program design. This will include:
 - a. Data Structure Design (file design with full description of the data, its type, its size, its purpose and any validation – if required).
 - b. Input Design (hand drawn or screen dumps of Forms where the data is to be entered).
 - c. Output Design (hand drawn or screen dumps of desired report format).
 - d. Detailed process description/diagram showing program logic/flow (you can use flow charts or process diagrams).

Program Development

[28 marks]

(a) Implementing the Program

[6 marks]

1. Include full program listing (excluding any third party control/library code) with full annotation where necessary.
2. Description of all procedures, functions and/or modules with short and meaningful annotations.
3. The program should be related to the design in functionality.

(b) Using Good Programming Style

[6 marks]

See "Guidelines for Project Software" section.

(c) Programming Skills

[16 marks]

Students must demonstrate the following skills and must identify the section of code where they have been used (you can use a highlighter and annotations):

1. arrays and/or records
2. different data types
3. selection
4. iteration
5. procedure
6. function
7. searching technique
8. file

You will be awarded 1 mark for a valid use of each of these techniques, and 1 mark for correctly annotating it in the code.

Testing

[8 marks]

1. For full marks, you should produce a Test Plan for each type of field. This test plan should include valid, extreme and invalid data along with their expected results and the actual results produced by the system.
2. Each test conducted must be supported by the evidence of the test. This could be done by using screen shots of the test.
3. Each test result must be annotated.

Implementation

[8 marks]

(a) Technical Documentation

[4 marks]

The technical documentation must be produced as a standalone guide for anyone who may be required to make modifications to the system. The guide should include the following sections:

1. Functional, Software and Hardware Requirements
2. Data Structure
3. Record/File dump
4. Data Dictionary
5. Data Flow Diagram
6. Input/Output screen dumps
7. Program Listing (annotated)
8. Flow Charts
9. Formulae used/Validation Checks

(b) Installation Instructions

[2 marks]

1. This should be a comprehensive and well illustrated guide to installing the program for use.
 2. A concise Glossary could be included for easy reference.
-
-