Microsoft
Visual Basic
6.0
1. Introduction

Visual Basic implements graphical user interface that allows the use of graphics for different applications. It provides visual interactive windows with user, like Dialogue box for (color, font ...), Input box, and Output box. Also it is able to create menu to simplify user application.

To run this program on user computer:
Start>programs>Microsoft Visual Studio 6.0>Microsoft Visual Basic 6.0.

It will appear on the computer screen as in the following picture.

To exit from Visual Basic and return to Windows is like exit from most Windows applications. There are three ways to close the Visual Basic as stated below.

1- Click on close button icon that appears in the upper-left corner of the screen.
2- Press Alt+F4
3- Select File >Exit

1.2-The Importance of Visual Basic Program

Languages like Basic and Pascal depend on variables and procedures to build the applications. This is why it is called procedural languages. The new approach is called object programming for visual programs like Visual Basic and Visual C++ and others. In this programming approach every thing (form, command buttons, controls) is an object.

The reasons for of implementing Visual Basic program are listed as follows:
1- It uses integrated development environment (IDE) which is easier for the user to minimize code writing.
2- All visual programs follow the same concepts, therefore the user will become more familiar with visual approach for other visual languages.

3- It provides Input box and Output box as an interactive windows with user.

4- It is able to connect to Internet, and to call Explorer.

1.3- Elements of the Integrated Development Environment (IDE)

The IDE environment consists of many elements. Some elements are displayed when Visual Basic is started (By default) as in the following figure. Other elements are displayed if the user requires them. We will list some of these elements.

- **a- Menu Bar**: It contains a standard command like: File, Edit, View, Window, Help menus, and specific command such as: Project, Format, or Debug menus.

- **b- Toolbar**: it contains the most commonly used commands (button), if clicked an action represented by that button is carried out.

- **c- ToolBox**: it contains a collection of tools that are needed for project design.
d- form Designer: it is a window for each form to customize the designed interface of the application. Using the form designer, the user can add controls, graphics, and text to create the desired form appearance.

e- Properties Window: it is a List of properties settings for a selected form or a control. These properties are characteristics (such as size, visible, or color) of the selected object it provides an easy way to set properties.

f- Project Explorer Window: it is a list of the forms and modules for the current projects. It is a hierarchical tree-branch structure, where the project at top of tree and other parts like forms, modules) descend from this tree.

g- Form Layout Window: The Form Layout window is a small screen. Which is used to reposition the form of the application so that it appears in proper place when project is run.

h- Code Editor Window: Code Editor Window is used to write a VB code for an application. For each form there is a separate code editor window. It is displayed when user clicks on form or object in form.

To Create an Application

The title of program includes the name of project, and when the user first starts the program it takes a defaulted value (project1). It also includes resize
icons. The following steps are required to create an application in Visual Basic 6.0:

1. Select type of project New or Exciting. A form automatically appears in the form design. The basis for any application's interface is the form that user should create. User can add other forms to the project (to add another form select project menu>add form).
2. To add objects (controls) to the form use the ToolBox.
3. Set the properties for the objects through properties window.
4. Write code. The Visual Basic Code consists of statements, and declarations.
   The code for an application can be written on the Code Editor window. In this window user can view and edit quickly any of the code.
5. Run the Application. To run the application, click the Start button on the toolbar, or press F5.
6. Stop. To stop running the application and return to visual basic program click on stop button in tool bar.
7. Check if there is an error, return to step 3 ,otherwise continue.
8. Save project.

**Exercises**

1. Define the following parameters: a-Toolbar.
   b-ToolBox. c-
   Form designer, d-
   form layout. e-
   project window,
   f-properties window. i-
   code editor window..

2. Explain the advantage of the Visual Basic program.
**Project**

Project is a program designed to user application that may be simple (like calculator program) or complex (like word program). Visual basic program can create many types of projects. The most important or usual project is the standard project (for window applications) and the DHTML project (for internet).

**Working with Standard Projects:**

The following working steps (create, save, add, open and delete) could be done:

a) To create project:

When program starts, project box appears-select Standard EXE > Project window appears.
OR: File> New project> Box (select Standard EXE)> Project window appears

b) To add project: Any number can be added.
Project icon> Select Standard EXE> Project window appears.

Note: Usually first project runs first, but user can change that by:
Selecting project from project window > mouse list > Set as startup.

c) To open an existing project:
It is previously designed and saved on disc in a folder.
File> Open project> Box (select existing and look for the project) > Project window

d) To delete a project:
Select project in Project window > Mouse list > Remove project.
e) To save project:
The visual basic can save the project on disc in two ways, as an executable type or a non-executable type.

I- for project in non execution stage:
There are many types of files summarized as follows:

1- Project file: it consists of all files which are related to specific project, also some other information with it. This could be saved with extension (.VBP)

2- The form Files: this contains form description and any Object or program related to it. This is saved with extension (.frm).

**To save project for first time:**
File>Save project (group) as>Box (project name)> forms saved then projects group saved.

**To resave project:** to save previously saved project in same place
File>save project (group)

**Note:** If a form is modified it should be saved. To save a form:
Select a form from Project window>File>Save project form1 as > Save box (select form name). OR: File>Save project forml.

II- project for execution: This is the final stage so that it could be opened and run by Windows and no need for Visual Basic program. File> Make project.exe.

<table>
<thead>
<tr>
<th>Item</th>
<th>Action steps</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create project</td>
<td>New</td>
<td>File&gt;New project</td>
</tr>
<tr>
<td></td>
<td>Exist</td>
<td>File&gt;Open project</td>
</tr>
<tr>
<td></td>
<td>Recent</td>
<td>File&gt;Open project</td>
</tr>
<tr>
<td>Save project</td>
<td>File&gt;Save project group as</td>
<td>Visual Basic can deal with it (open and modify).</td>
</tr>
<tr>
<td></td>
<td>File&gt;Make projectl.exe</td>
<td>For execution by window.</td>
</tr>
<tr>
<td>Delete project</td>
<td>File&gt; Remove project</td>
<td>Select project before remove.</td>
</tr>
</tbody>
</table>
Forms
1) Introduction to form
The form is the most important visible object, without it no control can be displayed. It is a window that can be designed and modified to fit user applications. In the standard project the form Designer creates and modifies visual forms. When user starts visual Basic program a form is automatically displayed in Designer window. The designer can add any number of forms to the project of his application by pressing: add form from project menu.

![Forms](image)

There are two modes: design mode and running mode. User can interchange between them, by pressing on start icon or stop icon on tool bar.

![Forms](image2)

The forms also have properties and events.

2) Form properties
Properties list has a predefined value (numeric or string) and could be changed, some properties could be rewritten like caption, and some could be selected from option list by pressing on down arrow on the side.

Others could be rewritten or by browsing the computer files when the user clicks on the dotted button on the right side a dialogue box.
appears. The browsing button appears when the user clicks inside the box.

The most important properties of the form are listed in the following table:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective</th>
<th>code</th>
<th>Stage of Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Used to represent name of form in code</td>
<td>Form\text sub{no}.caption = &quot;any name&quot;</td>
<td>Design</td>
</tr>
<tr>
<td>Caption</td>
<td>String appear in title of form</td>
<td>Form\text sub{no}.caption= &quot;any name&quot;</td>
<td>Design and run</td>
</tr>
<tr>
<td>Backcolor</td>
<td>Background color for form.</td>
<td>Form\text sub{no}.BackColor = Qbcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Forecolor</td>
<td>Color of text written on form.</td>
<td>Form\text sub{no}.ForeColor = Qbcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Font</td>
<td>Font style, type and size.</td>
<td>Size: Form\text sub{no}.FontSize = no. Style: \textit{italic} \textbf{bold} \underline{underline} Type: Form1.FontName = &quot;arial&quot;</td>
<td>Design and run</td>
</tr>
<tr>
<td>borderstyle</td>
<td>0 - None borderless and captionless 1 - Fixed Single a nonresizable form 2 - Sizable (default), creates a resizable window, 3 - Fixed Dialog: nonresizable form without Minimize and Maximize buttons 4 - Fixed Tool window for a floating toolbox like form, 5 - Sizable Tool window</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>Enabled</td>
<td>The tools enable or disable.</td>
<td>Form\text sub{no}.Enabled = true or false</td>
<td>Design and run</td>
</tr>
<tr>
<td>Min button Max button</td>
<td>=true. The Minimize and Maximize buttons are enabled. =false. The Minimize and Maximize button are disabled.</td>
<td>Form1.MaxButton = True or false \text{ = false} Form1.MinButton = True or false \text{ = false}</td>
<td>Run</td>
</tr>
<tr>
<td>Start up position</td>
<td>0 - Manual, use form layer window to position Form 1 - Center owner 2 - at Center Screen 3 - Windowdefault.</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>movable</td>
<td>True or false to make form movable or unmovable</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>Hide</td>
<td>To hide the form</td>
<td>Form\text sub{no}.Hide</td>
<td>Run</td>
</tr>
<tr>
<td>show</td>
<td>To show the form</td>
<td>Form\text sub{no}.Show</td>
<td>Run</td>
</tr>
<tr>
<td>icon</td>
<td>Change the icon on title bar of form (the icon must have the extension ico or cur)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3) Code form

The code is written in code Form and it will be edited quickly by code editor. The codes are of two categories:
1- Declaration is written before any procedure in the code
2- Statements. The user selects the required event then code statements are written inside these event procedures.

5) Events:

Events are like electrical switches. The electrical switches are of many types, so are the events.

The forms and controls support events (generation, interaction with mouse and keyboard). The most important events for the form are described in the following table.

<table>
<thead>
<tr>
<th>Event</th>
<th>Action taken when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click</td>
<td>Single click on object.</td>
</tr>
<tr>
<td>DbClick</td>
<td>Double click on object.</td>
</tr>
<tr>
<td>load</td>
<td>Loading the object</td>
</tr>
</tbody>
</table>

Examples:
1- Design a form such that: in event load, when project runs, the form backcolor property changed (chose any color).

sol:

```
Private Sub Form_Load()
    Form1.BackColor = QBColor(12) End Sub
```

2- Design a form such that: in event click on form, when project runs, the title of the form changed to applied science.

```
Private Sub Form_Click()
    Me.Text = "applied science" End Sub
```
Exercises:
Q1/ Design a standard project has three forms with backcolors red, blue and green.
Q2/ design a standard project that has one form change the name from form1 to "students"
Q3/ Create a standard project with one form the name of the project "market seals" and the name of the form "stock markets" and save project on desktop with name marketing .exe
Q4/ Design a standard project with four forms when running project all forms appear on screen Note :use this code in form1 (form1.show, form2.show, form3.show, form4.show)
Toolbox

It is a window box that contains tools which could be used in the project. Tools are objects that could be selected from Toolbox to be placed on form. To show the toolbox, Press toolbox icon > the toolbox appear as in the following diagram.

The toolbox includes many tools and in general they are:

1) Pointer (not control) used to select tools already on form
2) Picture box : Used to display images in any of the following formats: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, and JPEG.
3) Label : Fixed text appears on form for remark.
4) Textbox : For text edit. Like note pad.
5) Frame : To group tools together (container).
6) Command button : Used as a switch (such as OK and Cancel) buttons. Code is written in the Click event procedure of this control
7) Check box : For a yes/ no (true/false) selection.
8) Option button : For selection as group. Many options are placed inside container (grouped) (a Frame control). One control is selected from the group all others of the group are automatically deselected.
9) Combo box consists of (list and arrow when clicked a small a list appears), if user selects item from the list, it will be displayed in TextBox. Vertical size is fixed.
10) List box : For a list, user adds to and deletes from this list. It takes any size.
11) Horizontal Scrollbar : Create stand-alone Horizontal scroll bars.
12) Vertical Scrollbar : Create stand-alone vertical scroll bars.
13) Timer : Used to control object movement.
14) Drive List Box : It is a special ListBox filled automatically with names, of the files in a specified directory. It is a list invariant.
15) Dir List Box : It is a special ListBox filled with drives (Hard disc, Flopy, CD) in the system. It is an invariant.
16) File List Box : It is a special ListBox filled automatically with the names of all DirListBox. It is a list invariant.
17) Shape : Used only to display rectangles, circles, and ovals on the forms. Never raises any events
18) Line : Used only to display lines on the forms. It never raises any events.
19) Image : Used instead of PictureBox because it consumes fewer system resources.
20) Data : used for data base.
21) OLE : used for joining with another programs.
Tool Box and Form

The user can place the tool on form and then work with the tool. To place the tool on form:
Click on tool > Draw tool to Form > the tool appears on Form.
Or: double click on it.

Notes:
a) Each tool has a property window. To see this window: Click on tool on form > Property window appears.
b) Property can be changed manually or by code and the effect of code appears in the run time (when user runs project).
c) To put code for tool action:
Double click on tool > code sheet of the Form appears (with code of corresponding tool is written) > User write the desired code inside tool event, or outside in Form event.

Working With Tools

The user can work with tool in the design stage.
- To add tool: double click on tool. Tool appears on form or drags it to design part of page and draw it in the desired size.
- To delete: click on element in page > press delete key of the key board or right click on object for mouse list > choose delete.
- To display tool properties window: click on element > properties window appear.
- To display code form: double click on tool code form for that element.

Label:

It is used to display fixed text on form

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective</th>
<th>Code</th>
<th>Stage of Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>String appear on label</td>
<td>label_no_.caption= &quot;any name&quot;</td>
<td>Design and run</td>
</tr>
<tr>
<td>Autosize</td>
<td>To resize tool to fit text</td>
<td>label_no_.autosize= true or false</td>
<td>Design and run</td>
</tr>
<tr>
<td>Backcolor</td>
<td>Background color for label</td>
<td>label_no_.BackColor=QBcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Forecolor</td>
<td>Color of text written on label</td>
<td>label_no_.ForeColor=QBcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Font</td>
<td>Font style, type and size</td>
<td>Size: label_no_.FontSize= no. italic, bold, underline</td>
<td>Design and run</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type: label.FontName = &quot;arial&quot;</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>The label appear or disappear</td>
<td>Label_no_.Visible= true or false</td>
<td>Design and run</td>
</tr>
<tr>
<td>Enabled</td>
<td>The label enable or disable</td>
<td>label_no_.Enabled= true or false</td>
<td>Design and run</td>
</tr>
</tbody>
</table>

Note: The available color numbers that used with QBcolor is the integers 0 to 15 only.
Example: Design a form contains label "العلوم التطبيقية" in size 14.

Sol: the properties are:

<table>
<thead>
<tr>
<th>Label1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>caption</td>
<td>Applied science</td>
<td></td>
</tr>
<tr>
<td>fontsize</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

Running stage:

Textbox

The textbox is a box for entering and displaying text (characters or values) in user project. This tool is used frequently in most of the application. The textbox has property window, with no caption, but with space for text. The most important property of this tool is the text content which is described in the following:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective</th>
<th>Code</th>
<th>Stage of Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>String appear on textbox</td>
<td>text_no.text = &quot;any name&quot;</td>
<td>Design and run</td>
</tr>
<tr>
<td>multiline</td>
<td>To enter more than one line</td>
<td>true or false</td>
<td>Design</td>
</tr>
<tr>
<td>Backcolor</td>
<td>Background color for textbox</td>
<td>text_no.Backcolor=Qbcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Forecolor</td>
<td>Color of text written on textbox</td>
<td>text_no.forecolor=Qbcolor(no.)</td>
<td>Design and run</td>
</tr>
<tr>
<td>Font</td>
<td>Font style, type and size.</td>
<td>Size: text \textfontsize = \textno. \textitalic \textbold \textunderline \textstyle = \textno. \textitalic \textbold \textunderline \Type: \textname = &quot;\textarial&quot;</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>The textbox appear or disappear</td>
<td>text \textno. \textvisible = \texttrue or \textfalse</td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td>The textbox enable or disable.</td>
<td>text \textno. \textEnabled = \texttrue or \textfalse</td>
<td></td>
</tr>
<tr>
<td>passwordchar</td>
<td>A row of symbols appear instead of letters</td>
<td>Text \textno. \textpasswordchar = (\symbol)</td>
<td></td>
</tr>
<tr>
<td>Setfocus</td>
<td>Put the focus on the specified textbox</td>
<td>Text \textno. \textsetfocus</td>
<td></td>
</tr>
</tbody>
</table>

**Change text manually:** change text property from property window, click inside textbox and add text.

**Example:** Design a form to enter username and password such that the title of the form is VB.

**Sol:** design stage

```
<table>
<thead>
<tr>
<th>Form1</th>
</tr>
</thead>
<tbody>
<tr>
<td>caption</td>
</tr>
<tr>
<td>Text1</td>
</tr>
<tr>
<td>text</td>
</tr>
<tr>
<td>Text2</td>
</tr>
<tr>
<td>Text</td>
</tr>
<tr>
<td>Label1</td>
</tr>
<tr>
<td>caption</td>
</tr>
<tr>
<td>Label2</td>
</tr>
<tr>
<td>caption</td>
</tr>
</tbody>
</table>
```
**Example:** Design a form with one textbox, set the text properties so that this massage appears when project runs (welcome to visual basic world).

**Sol:** There are two methods:

**First method:** changing property by code:

Private Sub Form_Load()
Text1.Text = "welcome to visual basic world"
End Sub

**Second method:** by properties window

<table>
<thead>
<tr>
<th>Text1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
</tr>
</tbody>
</table>

Running stage

**Command button**

It acts as a switch. To deal with tool property> click on command button> property window appear> change setting of any desired property. Usually change set its caption property to a suitable string.

To make the button functional, the user should add some code. To do this: click on command tool> code form appears with click event procedure. Write code in this event or other events like press key event.
The most familiar properties that are needed for the command button are stated in the table below.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective</th>
<th>Code</th>
<th>Stage of Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>String appear on command</td>
<td><code>command_no.caption=&quot;any name&quot;</code></td>
<td>Design and run</td>
</tr>
<tr>
<td>Style</td>
<td>Determine the style of command</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>Backcolor</td>
<td>Background color for command</td>
<td></td>
<td>Design and run</td>
</tr>
<tr>
<td>Forecolor</td>
<td>Color of text written on command</td>
<td></td>
<td>Design and run</td>
</tr>
<tr>
<td>Font</td>
<td>Font style, type and size</td>
<td></td>
<td>Design and run</td>
</tr>
<tr>
<td>Visible</td>
<td>The command appear or disappear</td>
<td></td>
<td>Design and run</td>
</tr>
<tr>
<td>Enabled</td>
<td>The command enable or disable.</td>
<td></td>
<td>Design and run</td>
</tr>
</tbody>
</table>

**Example:** Design a form with label, such that when click on the command button "name" your name appears on label (at running stage).

**sol:**

```vbnet
Private Sub Command1_Click()
    Label1.Caption = "Ahmad"
End Sub
```

**Example:** Design a form to appear your name and department in textbox, when click on command button "name" and "department" respectively so that you can clear these informations when click on command "clear" and stop project when click on command "exit".
Private Sub Command1_Click()
    Text1.text="Muna"
End Sub

Private Sub Command2_Click()
    Text1.text="Science"
End Sub

Private Sub Command3_Click()
    Text1.text=" 
End Sub

Private Sub Command4_Click()
End Sub

Example: Design a form contains two textbox so that when click on command button "copy" the text copied from first textbox to the second textbox but in size (28).

Sol:

<table>
<thead>
<tr>
<th>Text1</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>caption</td>
<td>name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text2</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>caption</td>
<td>copy</td>
</tr>
</tbody>
</table>
Private Sub Command1_Click()
Text2.Text = Text1.Text
Text2.FontSize = 28
End Sub

At run stage this window appear

If the user enter by example the text (ABC) in first textbox and click on command (copy) the same text appear on the second textbox but in size 28.

Exercise:
1- Design a form to display your name in specific color and size.
2- Write the following design steps with details
**Shape:** Shape is a tool used to draw geometric shape (circle, rectangle, square, etc). It has property window. **It has no events like other tools** (such as click, dblclick, etc.).

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective</th>
<th>Stage of Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td>shape</td>
<td>To determine a specific shape: 0-rectangle 1-square 2-oval 3-circle 4-rounded rectangle 5-rounded square</td>
<td>Design and run</td>
</tr>
<tr>
<td>backstyle</td>
<td>0-trancsparence 1-opaque</td>
<td>Design</td>
</tr>
<tr>
<td>Backcolor</td>
<td>colored the Background for shape, appear after backstyle changed to value 1-opaque</td>
<td>Design and run</td>
</tr>
<tr>
<td>bordercolor</td>
<td>Colored the Border of shape</td>
<td>Design and run</td>
</tr>
</tbody>
</table>

**Example:** Design a form contains a specific shape then write the name of this shape on form.

**List box:** The user can't write directly in ListBox. He can add item to the ListBox property or by code in the form.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective and code</th>
</tr>
</thead>
<tbody>
<tr>
<td>sorted</td>
<td>True , to sort the elements alphabetically False , elements without sort.</td>
</tr>
<tr>
<td>Style</td>
<td>To determine the style of list: 0-standard 1-checkbox</td>
</tr>
<tr>
<td>Clear list</td>
<td>To clear all elements of the list: Listno.Clear</td>
</tr>
</tbody>
</table>

**Add items to list:**
a) Change property list from properties window. When click on arrow, write items (elements).
b) Add elements by code using the property additem as follows:

```
ListNo.additem ("first element ")
ListNo.additem ("second element ")
.
.
ListNo.additem ("last element")
```

**Example:** Design a form contains label to display your department and two command buttons "show" and "hide" such that when click on command1, form2 appears and when click on command2, form2 disappears. In form2 design a list to contain the name of departments branches which appears after click on command button "display".

**Form1**

- Label1: caption="applied science"
- Command1: caption="show"
- Command2: caption="hide"

**Form2**

- Command1: caption="display"
- List1: list=غرارة

```
Private Sub Command1_Click()
Form2.Show
End Sub
Private Sub Command2_Click()
Form2.Hide
End Sub
```
**Example:** Design a form contains a sorted list alphabetically such that the user can add the item from text to the list after click on command button "add".

**Sol:**

```vbnet
Private Sub Command1_Click()
    list1.AddItem (Text1.Text)
    Text1.Text = ""
End Sub
```

**Option button:** Used only as a group of buttons. When the user selects one of them the others are deselected automatically.

All other properties of this control are similar to those in form and command button where they are fully discussed which are caption, font, enabled, backcolor and visible beside an important property which is value that takes true or false and it used with if statement. The option button usually takes click event.

**Example:** Design a form with three option buttons " red ", " green " and " blue " such that when we click on options the color of the form colored by red, green and blue respectively.

```vbnet
Private Sub Option1_Click()
    Form1.BackColor = vbGreen
End Sub

Private Sub Option2_Click()
    Form1.BackColor = vbBlue
End Sub

Private Sub Option3_Click()
    Form1.BackColor = vbRed
End Sub
```
Check box:

Any number of check boxes can be used on a form. They work independently. Its Property value could be changed in design stage manually, or in running stage by code.

Example: Design a form with one text box and three check boxes such that when click on boxes the following is done: change typing to bold, italic, underline. Sol:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Text1</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ﻣ(150,414),(249,442)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Check1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>caption</td>
<td>Bold</td>
</tr>
<tr>
<td></td>
<td>Check2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>caption</td>
<td>Italic</td>
</tr>
<tr>
<td></td>
<td>Check3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>caption</td>
<td>underline</td>
</tr>
</tbody>
</table>

Private Sub Check1_Click()
Text1.FontBold = Check1.Value
End Sub
Private Sub Check2_Click()
Text1.FontItalic = Check2.Value
End Sub
Private Sub Check3_Click()
Text1.FontUnderline = Check3.Value
End Sub

Run stage:
**Timer**

Timer returns the time in millisecond. It may be used to measure execution time of code (program efficiency).

<table>
<thead>
<tr>
<th>Property name</th>
<th>Objective and code</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval</td>
<td>To repeat the code according to event. It takes an integer values (0-65535) and measured in millisecond.</td>
</tr>
<tr>
<td>enabled</td>
<td>timer=no. Enabled =true or false</td>
</tr>
</tbody>
</table>

Ex: design electronic clock to display the time in seconds.

sol:

```vbnet
Private Sub Timer1_Timer()
Label1.Caption = Time
End Sub
```

**Example:** Design a form to display "applied science" such that when click on command button "start" the color of "applied science" changed randomly every second.

Sol:

```vbnet
Private Sub Command1_Click()
Timer1.Enabled = True
End Sub
Private Sub Timer1_Timer()
t = Rnd * 15
Label1.ForeColor = QBColor(CInt(t))
End Sub
```
End Sub

Run stage:
When click on command button the color of the font will be changed every second randomly in integer no. (0-15).

Note: the function (Cint) used to convert to integer no.
And (Rnd) used to generate a random no. in a range (0-1)

**Input - output boxes**
There are two types of dialog boxes which are inputbox and messagebox. The first is used to input variable and the second to output variable or message. Both needs code and appear at run time.

a) Inputbox
Inputbox used to input value or characters for one variable from keyboard at running stage.
This box needs a code in code sheet and could be written in any event or command

X=Inputbox(" prompt or remark", "title")

**Example:** enter value of x using inputbox
Sol:

Private Sub Form_Load()
X=Inputbox("enter value of x", "calculation")
End Sub
Message box
It is used to output a message to the user (at running stage) the code needed could be written in code sheet and in any event or command.

The available icons for message box

<table>
<thead>
<tr>
<th>structure</th>
<th>value</th>
<th>icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbcritical</td>
<td>16</td>
<td><img src="image" alt="critical" /></td>
</tr>
<tr>
<td>vbquestion</td>
<td>32</td>
<td><img src="image" alt="question" /></td>
</tr>
<tr>
<td>vbexclamation</td>
<td>48</td>
<td><img src="image" alt="exclamation" /></td>
</tr>
<tr>
<td>vbinformation</td>
<td>64</td>
<td><img src="image" alt="information" /></td>
</tr>
</tbody>
</table>

The available commands for message box

<table>
<thead>
<tr>
<th>structure</th>
<th>value</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vbokonly</td>
<td>0</td>
<td>Ok</td>
</tr>
<tr>
<td>Vbokcancel</td>
<td>1</td>
<td>Ok, Cancel</td>
</tr>
<tr>
<td>vbAbortRetryIgnor</td>
<td>2</td>
<td>Abort, Retry, Ignore</td>
</tr>
<tr>
<td>vbYesNoCancel</td>
<td>3</td>
<td>Yes, No, Cancel</td>
</tr>
<tr>
<td>vbYesNo</td>
<td>4</td>
<td>Yes, No</td>
</tr>
<tr>
<td>vbRetryCancel</td>
<td>5</td>
<td>Retry, Cancel</td>
</tr>
</tbody>
</table>

For example if we write the following statement then a message box will be appear as shown below:

MsgBox "please close your program", 16, "Error"

or

MsgBox "please close your program", vbcritical, "Error"

Example: show what appear after running the following statement

MsgBox "are you sure you want to delete this file", 32 + 4, "delete"

or

MsgBox "are you sure you want to delete this file", vbQuestion+vbYesNo,"delete"

Sol:
Example: write a program to move the text (excellent) from textbox to message box and change the color of the text after click on command button (display).

Sol:

Text1: text="excellent"

Command1: caption="ضرع"

Private Sub Command1_Click()
    MsgBox (Text1.Text)
    Text1.BackColor = QBColor(9)
    Text1.Text = " "
End Sub

Or we can write the following code:

Private Sub Command1_Click()
    X = Text1.Text
    MsgBox (X)
    Text1.BackColor = QBColor(9)
    Text1.Text = " "
End Sub
Exercises
Q1/design a form contains specific title such that when we click on command1 the color of the font will change and we click on command2 the size of the font will be change.

Q2/design a form contains shape and command buttons "what is this" such that when click on the command button the name of this shape appears in message box.

Q3/enter a text in label1 such that this text copied to label2 after click on command button "copy" and delete from label2 after click on command button "delete".

Q4/design a form contain command button such that when we click on command, form2 will be appear and form1 will be disappear.

Q5/design a form "at run stage" the title of form1 will be change to "visual basic" and the back color of form1 will change too.

Q6/design a form with two shapes, red and blue, write code to replace the color every between two shapes.

Q7/design a form contain list box and two command buttons:
   1- add: to add the element to the list from input box
   2- delete: to delete the list elements
Visual basic statements
In visual basic program (code) there are four basic parts, i.e. it is contains the following statements:
  1- Declaration of variables and constants
  2- Inputting variables
  3- Operators for variables
  4- Outputting variables

1- **Declaration of a variable and constants**
The declaration means defining the data type (variable or constant).

**Variables**

A variable is a space in memory filled with data (value, character, time or date).

**Notes:**
- Variable name must start with character (not number or function) and maximum length 256 character, and does not contain point or symbol.
- Variable name must not repeat for other values.

The variable has to be declared. Variable type is defined by its content .The content may be data as numeric or character or string or Boolean or date, or any type of data (called variant), these types declared as:

**Dim** variable name **as** type  
**Or** 
**Global** variable name **as** type

**Note:** The **Dim** declaration written in general part of the form or in any place in form or sub procedure which used for one form. While **Global** declaration used for all forms

The types of variables that are allowed in visual basic are stated in the table below.

**Types of variables**

<table>
<thead>
<tr>
<th>Type</th>
<th>Value range</th>
<th>Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Integer</td>
<td>-32768&lt;x&lt;32768</td>
<td>Dim x as integer</td>
</tr>
<tr>
<td>2-Long</td>
<td>-2.1 e+009&lt;x&lt;2.1 e+009</td>
<td>Dim x as long</td>
</tr>
<tr>
<td>3-Single</td>
<td>1.4e-045 &lt;</td>
<td>x</td>
</tr>
<tr>
<td>4-Double</td>
<td>4.9e-324&lt;x&lt;1.79e+308</td>
<td>Dim x as double</td>
</tr>
<tr>
<td>5-String</td>
<td>65535 characters</td>
<td>Dim x as string</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>6-Boolean</td>
<td>True or false</td>
<td>Dim x as Boolean</td>
</tr>
<tr>
<td>7-Date</td>
<td>Computer time and date Jan 100&lt;x&lt; 31 Dec 9999</td>
<td>Dim x as date</td>
</tr>
</tbody>
</table>

**Constants**

It is a space in memory filled with fixed value that will not be changed. Constant may be declared as:

**Const** constant name = value

**Example:** Declare x as a constant (P), then compute the area of a circle. Put suitable design.

Sol:

![Image of a circle area application](Image)

<table>
<thead>
<tr>
<th>Form1</th>
<th>Area of a circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>caption</td>
<td></td>
</tr>
<tr>
<td>label1</td>
<td></td>
</tr>
<tr>
<td>Caption</td>
<td>radius</td>
</tr>
<tr>
<td>Text1</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>إَحْمَد</td>
</tr>
<tr>
<td>Command1</td>
<td>compute</td>
</tr>
<tr>
<td>Enabled</td>
<td>false</td>
</tr>
</tbody>
</table>

code stage:

```vbnet
Const p = 3.14159
Dim a, r As Single

Private Sub Text1_Change()
    Command1.Enabled = True
End Sub

Private Sub Command1_Click()
    r = Val(Text1.Text)
    a = r ^ 2 * p
    MsgBox("area=" & a)
    Text1.Text = " "
    Text1.SetFocus
End Sub
```
2- Inputting variables

There are methods to input variable x as stated in the following:

<table>
<thead>
<tr>
<th>Method of input</th>
<th>For all type of variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>In text tool</td>
<td>X=text_no,text</td>
</tr>
<tr>
<td>In input box</td>
<td>X=inputbox(&quot;prompt&quot;,&quot;title&quot;)</td>
</tr>
</tbody>
</table>

Note: To enter many variables we usually use the second method with loop.

3- Operators for variables

The operators that are used for variable are described in the following table

<table>
<thead>
<tr>
<th>Arithmetic operators</th>
<th>+</th>
<th>addition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>subtraction</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td></td>
<td>/</td>
<td>division</td>
</tr>
<tr>
<td>mod</td>
<td>Modulus – rest of division</td>
<td></td>
</tr>
<tr>
<td>^</td>
<td>exponent</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relational operators</th>
<th>=</th>
<th>equal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td></td>
<td>&lt;=</td>
<td>Less or equal</td>
</tr>
<tr>
<td></td>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td></td>
<td>&gt;=</td>
<td>Greater or equal</td>
</tr>
<tr>
<td></td>
<td>&lt;&gt;</td>
<td>Not equal</td>
</tr>
</tbody>
</table>

Note: The order of operations when executing arithmetic operation is:
Exponentiation - multiplication division and mod - finally addition and subtraction.

The mathematical representation must be written as visual basic representation in the code as following examples:

<table>
<thead>
<tr>
<th>Mathematical representation</th>
<th>Programming representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3(x+4y)</td>
<td>3*(x+4*y)</td>
</tr>
<tr>
<td>[ x^2 ]</td>
<td>[ X^2 ] \ 4 / 2</td>
</tr>
<tr>
<td>[ \sqrt[16]{3} ]</td>
<td>[ 16^{1/4} ] 10 5 4 3^2 2^3</td>
</tr>
<tr>
<td>[ \frac{5}{x^2} ]</td>
<td>[ (5*y)/(x^2-4)+x-1 ]</td>
</tr>
<tr>
<td>[ \frac{e^{2x}}{\cos(2x)} ]</td>
<td>[ Exp(2<em>x)/(\cos(2</em>x)+\sin(x)) ]</td>
</tr>
</tbody>
</table>
**Assignment statement**

There are many statements ways to fill a variable as follows: Variable = expression

Expression may include variables, operations and functions as follows:
1- Numerical variable. For example: i=3
2- Mathematical relation. For example: x=a/b
3- Characters variable (string). For example: t="abc"
4- Boolean variable (logical). For example: p=true

**Functions for variables**

The numeric and string variables are the most common used variables in programming, therefore V.B provides the user with many functions to be used with a variable to perform certain operations or type convention. The most common functions for numerical variable x

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs(x)</td>
<td>Absolute of x</td>
</tr>
<tr>
<td>Sqr(x)</td>
<td>Square root of x</td>
</tr>
<tr>
<td>Int(x)</td>
<td>Integer of x</td>
</tr>
<tr>
<td>Exp(x)</td>
<td>Exponential of x (e^x)</td>
</tr>
<tr>
<td>Fix(x)</td>
<td>Take the integer part</td>
</tr>
<tr>
<td>Sin(x), cos(x), tan(x)</td>
<td>Trigonometric functions</td>
</tr>
<tr>
<td>Log(x)</td>
<td>Natural logarithms</td>
</tr>
<tr>
<td>Len(x)</td>
<td>Number of character of variable x</td>
</tr>
<tr>
<td>Lcase(x)</td>
<td>Change the text x to small letters</td>
</tr>
<tr>
<td>Ucase(x)</td>
<td>Change the text x to capital letters</td>
</tr>
<tr>
<td>Cint(x)</td>
<td>Convert x to integer</td>
</tr>
<tr>
<td>Clong(x)</td>
<td>Convert x to long integer</td>
</tr>
<tr>
<td>Cdbl(x)</td>
<td>Convert x to double precision</td>
</tr>
<tr>
<td>Cstr(x)</td>
<td>Convert variable x to string</td>
</tr>
<tr>
<td>Val(x)</td>
<td>Convert string x to numerical variable</td>
</tr>
</tbody>
</table>

**Note:** the last five functions are called conversion functions.

The following functions for different x are given for comparison.

<table>
<thead>
<tr>
<th>Function</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X=lcase(&quot;MY NAME IS&quot;)</td>
<td>my name is</td>
</tr>
<tr>
<td>X=ucase(&quot;my name is&quot;)</td>
<td>MY NAME IS</td>
</tr>
<tr>
<td>int(2.5)</td>
<td>2</td>
</tr>
<tr>
<td>Int(-2.5)</td>
<td>-3</td>
</tr>
<tr>
<td>Fix(2.5)</td>
<td>2</td>
</tr>
<tr>
<td>Fix(-2.5)</td>
<td>-2</td>
</tr>
</tbody>
</table>
4- Outputting variables

There are methods to output variable x as stated in the following:

<table>
<thead>
<tr>
<th>Method of output</th>
<th>For all type of variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>On form</td>
<td>Print x</td>
</tr>
<tr>
<td></td>
<td>Note: in load event we must use the statement: (form1.show)</td>
</tr>
<tr>
<td>to text tool</td>
<td>text_no.text =X</td>
</tr>
<tr>
<td>to label tool</td>
<td>Label_no.caption=x</td>
</tr>
<tr>
<td>By message box</td>
<td>msgbox (x)</td>
</tr>
<tr>
<td></td>
<td>Or msgbox (&quot;remark&quot;&amp; x)</td>
</tr>
</tbody>
</table>

The instruction print could be very helpful to display data and used as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>print</td>
<td>To leave one line and print on next</td>
<td></td>
</tr>
<tr>
<td>Print &quot;a&quot;, &quot;b&quot;, &quot;c&quot;</td>
<td>Use (,) to print a distance between outputs</td>
<td>a b c</td>
</tr>
<tr>
<td>Print &quot;a&quot;; &quot;b&quot;; &quot;c&quot;</td>
<td>Use (;) to print the outputs adjacent</td>
<td>abc</td>
</tr>
<tr>
<td>Print &quot;a&quot;,&quot;b&quot;; Print &quot;c&quot;</td>
<td>Print a, b then print c on the same line</td>
<td>abc</td>
</tr>
</tbody>
</table>

Example 1: write a program to enter any text and compute its length. Put suitable design.
Sol:

Design stage:

```
Form1
Caption | String length
Command1
caption | Click here
Label1
Caption | Length=
Text1
Text
```
Code stage:
Dim s As String
Private Sub Command1_Click()
s = InputBox("inter string")
L = Len(s)
Text1.Text = CStr(L) End Sub

Running stage:

Example2: write a program to add and subtract two integer numbers after putting a suitable design. Use message box for outputting.

Design stage:
Code stage:
Dim x, y, z as integer
Private sub command1_click()
X=val(text1.text)
Y=val(text2.text)
Z=x + y
Msgbox("addition result="&z)
End sub

Private sub command2_click()
X=val(text1.text)
Y=val(text2.text)
Z=x - y
Msgbox("subtraction result="&z)
End sub

Running stage
Enter two values in text1 and text2. When click on command (+) or (-) the addition or subtraction result appears in message box.

Example 3: write a program to execute the four operations according to the following design.

Code stage:
Dim a, b, c as single
Private sub command1_click()
a=val(text1.text)
b=val(text2.text)
c=a + b
End sub
Private sub command2_click
() a=val(text1.text)
b=val(text2.text)
c=a - b
End sub

Private sub command3_click
() a=val(text1.text)
b=val(text2.text)
c=a * b
End sub

Private sub command4_click
() a=val(text1.text)
b=val(text2.text)
c=a / b
End sub

Private sub command5_click()
Text3.text=cstr(c)
End sub

Private sub command6_click()
Text1.text=" "
Text2.text=" "
Text3.text=" "
End sub

Example4: write a program to compute the functions: sine, cosine, integer value, square, absolute value.

sol:

Design stage:
Dim x, y As Single
Private Sub command1_click()
x = Val(Text1.Text)
y = Abs(x)
Text1.Text = CStr(y)
End Sub

Private Sub Command2_Click()
x = Val(Text1.Text)
y = Sqr(x)
Text1.Text = CStr(y)
End Sub
Private Sub Command3_Click()
    x = Val(Text1.Text)
    y = Int(x)
    Text1.Text = CStr(y)
End Sub

Private Sub Command4_Click()
    x = Val(Text1.Text)
    y = Sin(x * 3.14159 / 180)
    Text1.Text = CStr(y)
End Sub

Private Sub Command5_Click()
    x = Val(Text1.Text)
    y = Cos(x * 3.14159 / 180)
    Text1.Text = CStr(y)
End Sub

Running stage
Q1/ Write a Program to compute the area of triangle with design.

Q2/Write a program to enter the name of a student and two marks of any subject by input box then computes the average and display the name and average in two labels.

Q3/ Write a program to enter a real number then find its square after putting suitable design.

Q4/ Design a project with three forms, such that form1 contains two command buttons "Pethagors equation" and "area of triangle" such that when click on command1 "Pythagoras equation", form2 will be appear so in this form the third side of triangle will be compute with Pythagoras equation after enter the first and second side, and when click on "area of triangle" the third form displayed so that the area will be compute after entering the base and height. Use suitable output and input methods.

Q5/ Write a program to display the message below after click on command button "run".

Q6/Write a program to display the time and the date, put a suitable design.
**Menu**

The menu is a bar at the top of the form. The standard form is display without menu, but the user can add it. This menu could be included in form using menu editor. In next section the menu editor and the required code will be discussed.

**Menu Editor**

To use menu there are three ways:
1- Press menu icon from toolbar.
2- press (ctrl+E) from key board.
3- click on: tools>Menu Editor.

Menu editor box appears as shown below.

The steps of applying menu editor box are as follows:

<table>
<thead>
<tr>
<th>To add item</th>
<th>Use menu editor as follows</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Menu Editor" /></td>
<td><img src="image" alt="Menu Editor" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To add many items</th>
<th>Use the menu Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Menu Editor" /></td>
<td><img src="image" alt="Menu Editor" /></td>
</tr>
</tbody>
</table>
Create sub Menu list

Use menu Editor

Create Shortcut

Use & with caption and select shortcut

**Code for menu items**

Each item in menu or sub menu is considered as a command which takes the event click only. The user can add code for each item: click on item > code for that item appears on code sheet. Also code can be added to form: click on item > code for that item appears on code sheet. This is described in the following figure:

**Example:** Design a form with menu and a label with a specific title. The menu contains one item color with sub menu items: red, green, blue and exit, to color the label in red, green, blue then exit from the program.

Sol: put label1 with any caption for example (hello)

<table>
<thead>
<tr>
<th>Caption: color</th>
<th>Create standard menu (color) from menu editor&gt;next</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: command1</td>
<td></td>
</tr>
<tr>
<td>Caption: red</td>
<td>Add sub menu items by pressing ⌘ then enter the caption and name&gt;next.</td>
</tr>
<tr>
<td>Name: command2</td>
<td></td>
</tr>
</tbody>
</table>
To programming these commands click on each one to open its code window and write the following code:

Private sub command2_click ()
Label1.backcolor=vbred
End Sub

Private sub command3_click ()
Label1.backcolor=vbgreen
End Sub

Private sub command4_click ()
Label1.backcolor=vbblue
End Sub

Private sub command5_click ()
End
End Sub
Conditional statements
There are two types of conditional statements:
1- If statement
2- Select case

1- If statement: The comparison operations are used with conditional statements.
The comparison operations are: (<, <=, >, >=, =,<> , and, or)

There are four structures for if statement.
a) Simple structure If.. then:
   Used for running one programming statement only if the required condition satisfied.
The general form is: If condition then statement

Example 1: write a program to enter a mark of a student then print (pass) if he successful.
Sol:
   Dim x as integer
   Private sub command1_click()
   X= cint(text1.text)
   If x>= 50 then text2.text= "pass"
   End sub

b) If block structure: Used for running many programming statements if the required condition satisfied.
The general form is:
   If condition then
   Statements
   End if

Example 2: write a program to enter a mark of a student then print (pass) in size 18 if he successful.
Sol:
   Dim x as integer
   Private sub command1_click()
   X= cint(text1.text)
   If x>= 50 then
      text2.text= "pass"
      text2.fontsize=18
      end if
   End sub
c) **If.. Then.. Else** structure: Used for running many programming statements if the required condition satisfied. And running another programming statements (after else) if the required condition not satisfied.

   The general form is:
   
   ```
   If condition then
   Statements
   Else
   Statements
   End if
   ```

   **Example 3:** write a program to enter a mark of a student then print (pass) if he successful and print (fail) otherwise.

   **Sol:**
   
   ```
   Dim x As Integer
   Private Sub command1_click()
   x = CInt(Text1.Text)
   If x >= 50 Then
   Text2.Text = "pass" Else
   Text2.Text = "fail"
   End If
   End Sub
   ```

d) **If.. Then.. Elseif.. Else** structure:

   Used if we have many conditions to be satisfied

   **Example 4:** write a program to enter a user name and display the message (hello) three times. The first one for (Muna), the second one for (Maha) and the third for any user as a guest.

   **Sol:**
   
   ```
   Dim x As String
   Private Sub command1_click()
   x = Text1.Text
   If x = "Muna" Then
   MsgBox "hello, Muna"
   ElseIf x = "Maha" Then
   MsgBox "hello,Maha"
   Else
   MsgBox "hello, guest"
   End If
   End Sub
   ```
Example 5: Write a program to classify any entered number according to its sign and display the phrase (negative number) when the number is negative and the phrase (positive number) when the number is positive, otherwise display the phrase (neither positive nor negative).

Sol:

```vbnet
Dim x As Single
Private Sub command1_click ()
x = Val(Text1.Text)
If x > 0 Then
    MsgBox "positive number"
ElseIf x < 0 Then
    MsgBox "negative number"
Else
    MsgBox "neither positive nor negative"
End If
End Sub
```

**Nested If statement:**

It can be takes the following structure:

```
If condition then
    If condition then
        Statements
    End if
End if
```

**Note:** Any structure of if structures can be used instead of structure 1 and 2 above.

Example 6: Write a program to enter two numbers and compute multiplication and division operations using option button with display the phrase (illegal division operation) when the denominator is zero.

Sol:

```vbnet
Dim a, b, c As Single
Private Sub command1_click()
a = Val(Text1.Text)
b = Val(Text2.Text) If Option1.Value Then
    c = a * b
    Text3.Text = CStr(c)
Else
    MsgBox "illegal division operation"
End If
End Sub
```
If \( b \neq 0 \) Then
   \( c = \frac{a}{b} \)
   Text3.Text = CStr(c)
Else
   Text3.Text = "illegal division operation" End If
End If
End Sub

### Select statement

Used for applying many statements depending on one variable. The general form is:

```
Select case variable
   Case value1
      statements
   Case value2
      Statements
   .
   .
   Case value
      n Statements
   Case else
      Statements
End select
```

#### Example 7:
write a program to print the days of the week when we enter its number

**Sol:**

Dim x As Integer
Private Sub Command1_Click()
x = CInt(Text1.Text)
Select Case x
   Case 1
      MsgBox ("Sunday")
   Case 2
      MsgBox ("Monday")
   Case 3
      MsgBox ("Tuesday")
   Case 4
      MsgBox ("Thursday")
   Case 5
      MsgBox ("Wednesday")
End Select
```
Case 6
MsgBox ("Friday")
Case 7
MsgBox ("Saturday")
End Select
End Sub

**Example 8:** write a program to give the evaluation for different marks as follows:

<table>
<thead>
<tr>
<th>mark</th>
<th>evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>90-100</td>
<td>Excellent</td>
</tr>
<tr>
<td>80-89</td>
<td>Very good</td>
</tr>
<tr>
<td>70-79</td>
<td>Good</td>
</tr>
<tr>
<td>60-69</td>
<td>Medium</td>
</tr>
<tr>
<td>50-59</td>
<td>Pass</td>
</tr>
<tr>
<td>0-49</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Dim x As Integer
Private Sub Command1_Click()
x = CInt(Text1.Text)
Select Case x
Case 90 To 100
MsgBox ("excellent")
Case 80 To 89
MsgBox ("very good")
Case 70 To 79
MsgBox ("good")
Case 60 To 69
MsgBox ("medium")
Case 50 To 59
MsgBox ("pass") Case 0 To 49
MsgBox ("fail")
Case Else
MsgBox "the range is 0-100", vbCritical, "error"
End Select
End Sub
Exercises

Q.1/ Design a form with three menus and label with a specific title. The menus contain sub menus as follows: color: red, blue, green, style: underline, bold, italic, size: 10, 20, 30 then write the code for each command.

Q.2/ Write a program to enter a value of x and compute the value of y
\[ y = \begin{cases} x & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases} \]

Q.3/ Write a program to find the root of first degree equation.

Q.4/ Write a program to enter two numbers and print the largest number.

Q.5/ Write a program to enter two numbers and compute multiplication and division operations using check box with displaying the phrase (illegal division operation) when the denominator is zero.

Q.6/ Design a form with a text box. Use select statement so that when user enters g, b, r and y then form colored to green, blue, red, and yellow respectively.
Loop statement:
Visual basic supports statement to perform loops. The loops statements could have different structures as follows:
1- Counter loop.
2- Conditional loop.

1- Counter loop:
Loops apply programming statements for fixed number of times using counter (for… next) statement.

The general form is:
```
For variable = start value to end value step step value
   Statements
Next variable
```

Example1: Write a program to print (hello) five times.
**Sol:**
```
Dim i as integer
Private Sub Command1_Click()
   For i = 1 To 5
      Print "hello"
   Next i
End Sub
```

Example2: Write a program to print even numbers from 1 to 10.
**Sol:**
```
Dim i as integer
Private Sub Command1_Click()
   For i = 2 To 10 step 2
      Print i
   Next i
End Sub
```

Notes:
1- The variable's value that we use as counter must be integer value (integer, long).
2- If we don’t determined the step value then the assumed value is 1.
3- If the final value smallest than the initial value, then the step value must be negative.

2- Conditional loop
Loops repeat programming statements according to specific condition. There are two types of conditional loop:
   1- Do while
   2- Do until

1- **Do while loop:** In this loop the statements will be implemented and repeated when ever the condition satisfied. The general form is:
**Do while** condition
Statements
**Loop**

**Example 3:** Write a program to print (hello) five times with its numbering using do while loop.

**Sol:**
Dim i as integer
Private Sub Command1_Click
() i = 1
**Do while** i <=
5 Print "hello";
i i = i + 1
**Loop**
End Sub

**Example 4:** Write a program to print even numbers from 1 to 10.

**Sol:**
Dim i as integer
Private Sub Command1_Click ()
i = 2
**Do while** i <= 10
Print i
i = i + 2
**Loop**
End Sub

2-**Do until** loop: In this loop the statements will be implemented and repeated whenever the condition not satisfied, (i.e) the loop will be stopped when the condition satisfied. The general form is:

\[
\text{Do until } \text{condition} \\
\text{Statements} \\
\text{Loop}
\]

**Example 5:** Write a program to print (hello) five times with its numbering using do until loop.

**Sol:**
Dim i as integer
Private Sub Command1_Click
() i = 1
**Do until** i > 5
Print "hello";
i i = i + 1
**Loop**
End Sub
**Example6:** Write a program to find the summation of undetermined number of positive numbers such that the program will be stopped when we enter negative number.

**Sol:**

```vba
Dim x, sum As Single
Private Sub command1_click()
    sum = 0
    x = Val(InputBox("enter x", "summation")) Do While x >= 0
        sum = sum + x
        x = Val(InputBox("enter x", "summation"))
    Loop
    MsgBox (CStr(sum))
End Sub
```

**Example7:** Write a program to find the summation of the numbers from 5 to 15.

**Sol:**

```vba
Dim I, sum as integer
Private Sub command1_click()
    sum = 0
    For i = 5 to 15
        Sum = sum + i
    Next i
    Label1.caption = "sum =" & cstr(sum)
End Sub
```

**Example8:** Write a program to find the summation of 10 numbers.

**Sol:**

```vba
Dim i as integer
Dim x, sum as double
Private Sub command1_click()
    sum = 0
    For i = 1 to 10
        x = val(inputbox ("enter number")) Sum = sum + x
    Next i
    Label1.caption = "sum=" & cstr(sum)
End Sub
```

**Running stage:**

For example if we entered the numbers: 1, 5, -1, 3, 2, 0, -1, 3, 0, -4 then sum=8
Example 9: Write a program to find the average of n numbers.

Sol:
Dim i as integer
Dim x, sum, av as Double
Private Sub command1_click()
    i = 1: sum = 0
    n = cint(text1.text)
    Do while i <= n
        x = val(inputbox("enter number"))
        Sum = Sum + x
        i = i + 1
    Loop
    Av = sum/n
    Text2.text = cstr(av)
End Sub

Example 10: Write a program to print multipliers of 5 (from 5 to 50)

Sol:
Dim i as integer
Private Sub Command1_Click()
    i = 5
    Do until i > 50
        Print i
        i = i + 5
    Loop
End Sub

Example 11: Write a program to find the average of numbers that dividable by 3 (without remainder) from 3 to 99.

Sol:
Dim I, n, sum as integer
Dim av as Double
Private Sub command1_click()
    i = 3 : n = 0
    sum = 0
    Do while i <= 99
        Sum = Sum + i
        n = n + 1
    Loop
    Av = sum/n
    Print "av = ", av
End Sub
Example 12: write a program to print (welcome) ten times, the first one with the ordinary size and color. Then make the color changed and the size bigger at each time.

**Sol:**

```
Dim i As Integer
Private Sub Command1_Click()
    Print "welcome"
    For i = 1 To 9
        FontSize = 10 + i
        ForeColor = QBColor(i)
        Print "welcome"
    Next i
End Sub
```

Series:
To compute the value of series, we use suitable loop statements according to the boundaries (limits) of each series.

Example 13: Find
\[
\text{Sum} = 1 + x + x^2 + x^3 + \ldots + x^n ,
\]
where \( x \) is an integer.

**Sol:**

```
Dim I, n, x, sum as integer
Private Sub command1_click ()
    sum = 1
    n=cint(text1.text)
    x=val(text2.text)
    For i = 1 To n
        Sum = sum +x^i
    Next i
    Text3.text=cstr(sum)
End Sub
```

Nested loop:
The nested loops are the loops that are placed inside each other. The most inner loop will be executed first, then the outer ones. These loops should neither intersect, nor have the same index. As follows:

```
For i = 1 To n
    For j = 1 To m
        Statements
    Next j
Next i
```
Example 14: write a program to print the multiplication table.
Sol:
Dim I, j As Integer
Private Sub command1_click()
For I = 1 To 10
For j = 1 To 10
p = I * j
Print I; "*"; j; "="; p,
Next j
Print
Next I
End Sub

Example 15: write a program to generate the numbers in the following form.

```
1 1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Sol:
Dim I, j As Integer
Private Sub command1_click()
For I = 1 To 5
For j = 1 To i
Print j;
Next j
Print
Next I
End Sub
Exercises:
Q.1: write a program to find n factorial (n!)

Q.2: write a program to enter n numbers and find the average of negative numbers only.

Q.3: write a program to print the multipliers of 6 from 12 to 60, using do while first, then use do until loop secondly.

Q.4: compute the value of y where:

\[
y = \frac{1}{x^1} + \frac{2}{x^2} + \frac{3}{x^3} + \frac{n}{x^n}
\]

Q.5: Find the odd numbers from 1 to 100.

Q.6: Write a program to generate the numbers as follows:

5 5 5 5
4 4 4
3 3
2

Q.7: How many multipliers of 3 are there between 1 and 100? Write a program to find that.