DRAFT

INFORMATION AND COMMUNICATION TECHNOLOGY

LEARNING MODULE

PROGRAMMING

Pusat Perkembangan Kurikulum
Kementerian Pelajaran Malaysia
2006

1. What is it?

   This is a learning module for a specific Learning Outcome as stipulated in the HSP that is :

   - 5.2 Program Development

2. Who is it for?

   This module is for students who are taking the ICT subject as their elective and for those who are interested in ICT.

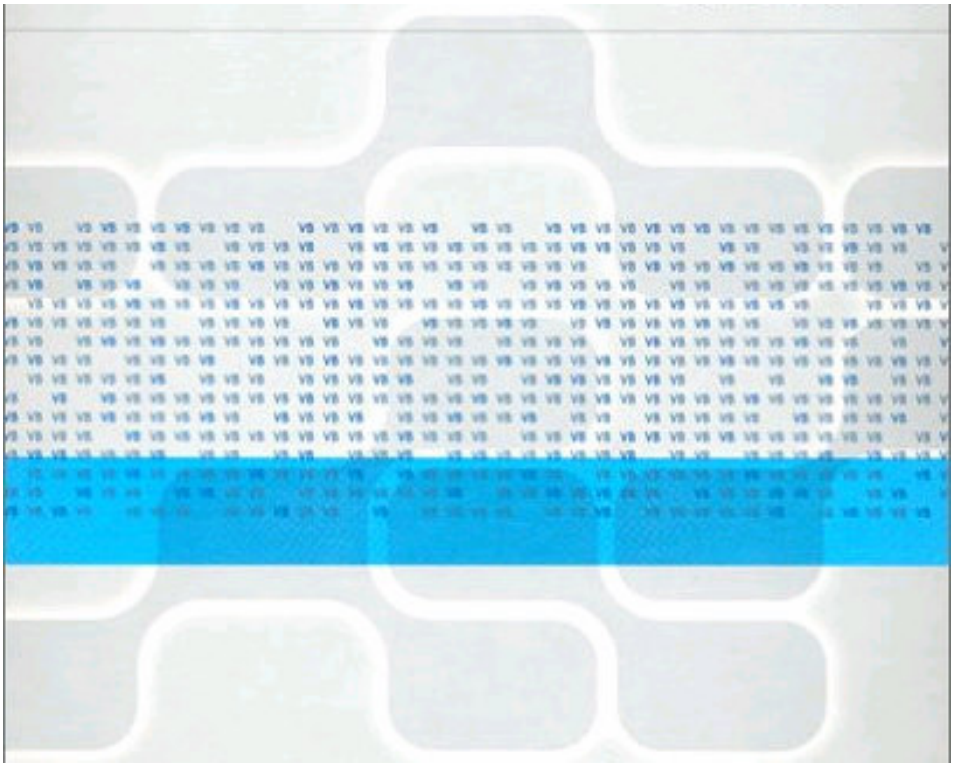
3. What can be achieved?

   Upon completion of this module, you should be able to :

   - Apply program development phases to solve problems

4. Is previous knowledge necessary?

   No. The content of this module will enable you to acquire the knowledge.



TOPIC : **Problem Analysis**

LEARNING OUTCOMES :

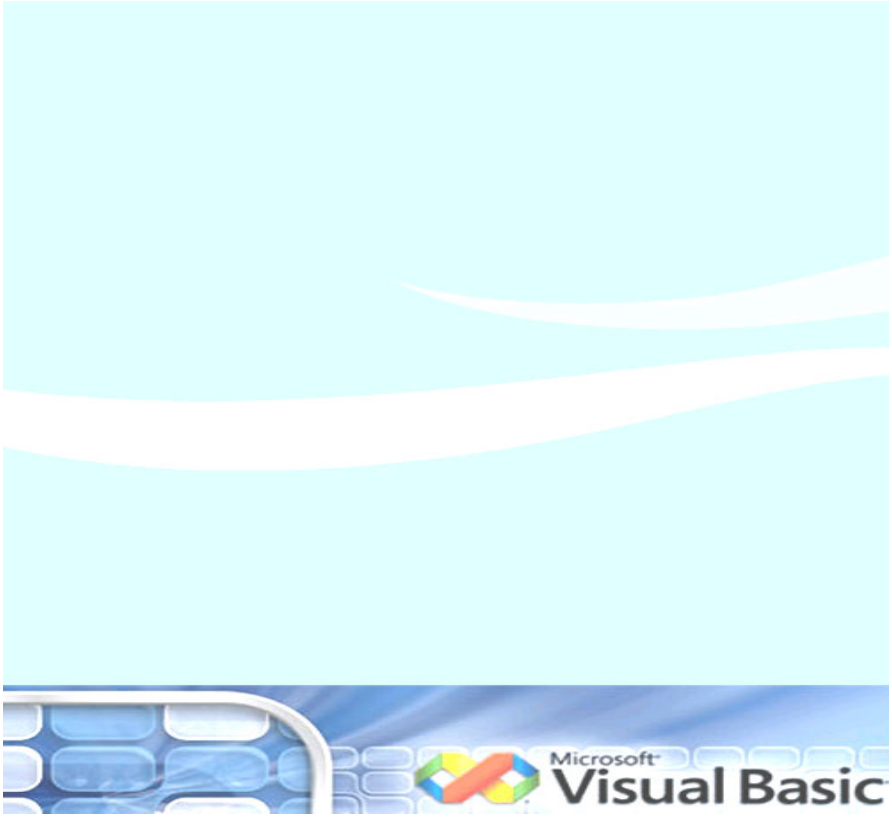
You should be able to:

1. Identify the problem,
2. Identify the input, processing and output.

DURATION : 2 periods

REFERENCES :

1. Schneider, D.I. (2004). An Introduction to Programming Using Visual Basic 6.0. Prentice Hall. New Jersey.
2. Behrouz A. Forouzan (2001). Computer Science, A Structured Programming Approach Using C, Brooks / Cole. California.
3. Timothy & Linda O'leary, (2007). Computing Essentials 2007, McGraw Hill, New York.
4. Garry B. Shelly (2007). Discovering Computers 2007. Thomson Course Technology, Massachusetts.



READING ACTIVITY

What is problem analysis?

We solve all kinds of problems in our daily life. A problem may involve how to commute to school, work or which homework to do first. Programming is the same. You need to make a clear statement of the problem you are trying to solve.

What should you do in this process?

To define a problem is the most difficult and important part of a programming project. Usually the problem can be divided into three components: input, process and output. Outputs are the goal of the problem solution. Inputs are the data you use to solve the problem. Processes are the steps needed to convert input information into the output results.

Based on the information given, read the situation below and define the problem, input, process and output.

Situation
Encik Malik is the manager of Enviro Car Enterprise. He asks you to develop a system to calculate the monthly instalment for every potential customer.

Encik Malik has made an appointment to see you at his office at 9.00 am on Thursday morning

You : Good morning, En. Malik.

En. Malik : Good morning. Ahh.. You are punctual. I like that.

You : Certainly. I appreciate doing business with you.

En. Malik : Have a sit. Let us discuss the problems that I'm facing.

You : Thank you.

En Malik : Now, where should we start?

You : Just tell me, what do you want me to do?

En. Malik : When my customer decides to buy a car, I will get the customer's details and fill it in a form. **My problem is, I have to calculate the monthly instalment manually.** It is very tiring as I always have to recalculate to make sure there are no mistake.

You : I see. Do you face any other problem?

En. Malik : Now the prices of cars has gone down and I foresee that the number of my customers is increasing. This phenomenon has increased my workload.

You : I see. You mentioned that you have to calculate the **monthly instalments**. Is there any other information that you need to give to your customer?

En. Malik : Usually I will give them the instalment details including the monthly balance every month until they complete their payments.

You : How do you determine a customers eligibility to get the loan?

En. Malik : I will check their **net income**.

You : What is the information you need to process the monthly instalment?

En. Malik : I need to know the **car model**, **car price, down payment**, **number of years** for the loan, and its related **annual interest percentage**.

You : How do you calculate the monthly instalment?

En. Malik : **The calculation for monthly instalment is quite complicated.** I have written down the calculation on this paper. Can you understand it?

Name _____

$$\text{Monthly Intalment} = \frac{\left(\left(\text{Loan} \times \left(\frac{\text{Annual Interest Percentage}}{100}\right)\right) \times \text{Number of Years} + \text{Loan}\right)}{\text{Number of Year} \times 12}$$

You : Yes, thank you. Another question, does your customer decide on the payment period?

En. Malik : Yes. If the customer is a government servant, we will let them pay either 10, 7 or 5 years. Otherwise the payment period will be 7 or 5 years only.

You : OK. Is there any other information that you need to tell me about your manual system?

En. Malik : That's all.

You : Alright, I'll discuss with my team members and will inform you then. Thank you.

En. Malik : You're welcome.

WRITING ACTIVITY

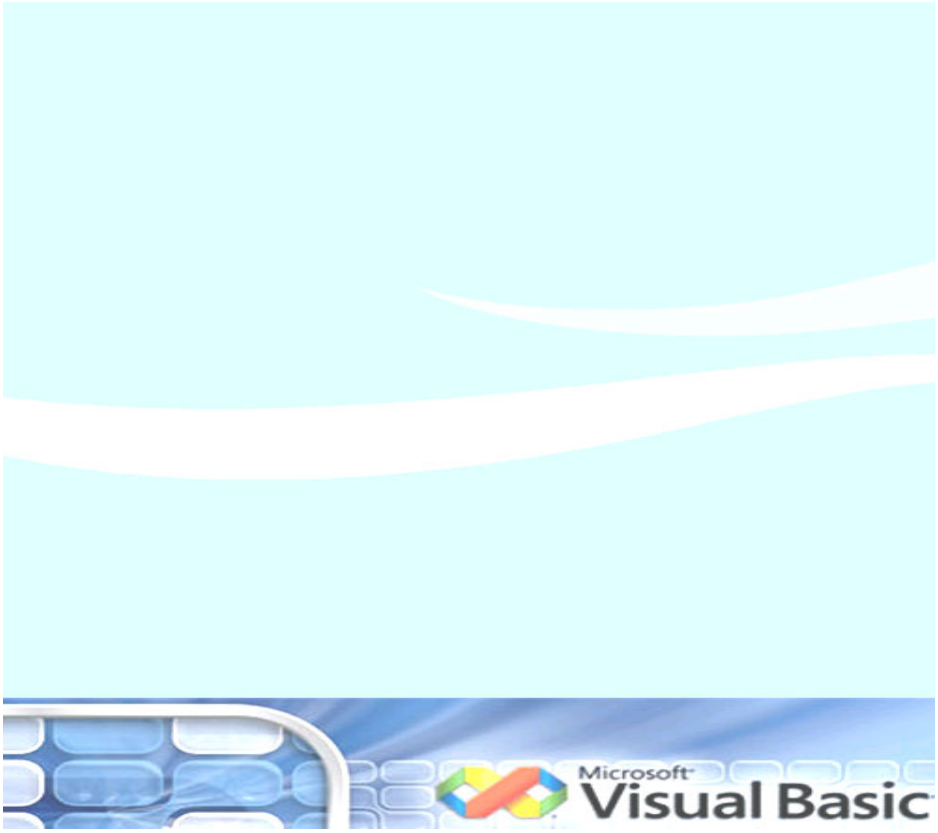
Based on the situation, answer the questions below.

1. Define the problem En. Malik faces in his company.

2. Identify the input En. Malik needs to collect everytime he meets a potential customer.

3. Identify the necessary output En. Malik sees in his manual system.

4. Write a formula on how to calculate the monthly instalment.



TOPIC : Program Design

LEARNING OUTCOMES:

By using this module, you will be able to create

1. Write pseudocode
2. Create flowchart
3. Design user interface

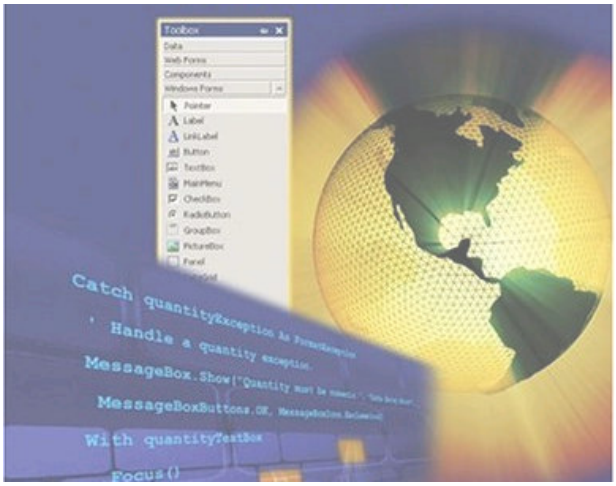
DURATION : 4 periods

READING ACTIVITY

Input Process Output (IPO) Chart

Before writing a pseudocode, it is always a good practice to draw an Input Process Output (IPO) chart. The IPO chart is a tool for collecting input, process and output. To use the IPO chart, you need to start with the outputs which is the goal of the program, then list the inputs, finally list the steps needed to convert the input information into the output you want. Here is the IPO chart that shows you the necessary input, process and output for En. Malik's new program.

Input	Processing	Output
1. Customer's name 2. Net income per month 3. Model of the car 4. Price of car 5. Down Payment 6. Annual Interest percentage 7. Number of Years depend either for Government (10, 7 or 5 years) or Private servant (7 or 5 years).	Loan = Price – Down Payment Monthly Instalment = (((Loan x (Annual Interest Percentage / 100)) x Number of Years + Loan) / (Number of Years x 12)	Monthly Instalment

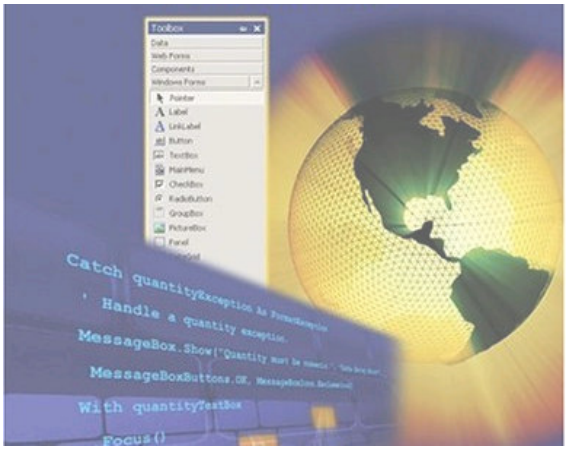


Pseudocode

Pseudocode is the English-like statements that follow a loosely defined syntax and are used to convey the design of an algorithm. It shows an outline of the logic of the program that programmers use to help them plan an algorithm. It is like writing the step-by-step summary of the program. It also states the process part of the IPO charts. A pseudocode always start with the word **PROGRAM** followed by the program name, and end the pseudocode with **END PROGRAM**. A sample of pseudocode based on the given situation, is shown below.

```
PROGRAM Calculate_Instalment

READ customer name

READ net income

CHECK the customer's net income

    READ the car model

    READ the car price

    READ down payment

    READ annual interest percentage

    READ the number of years for payment

    CALCULATE the monthly instalment

    PRINT the monthly instalment

END PROGRAM
```

Pseudocode

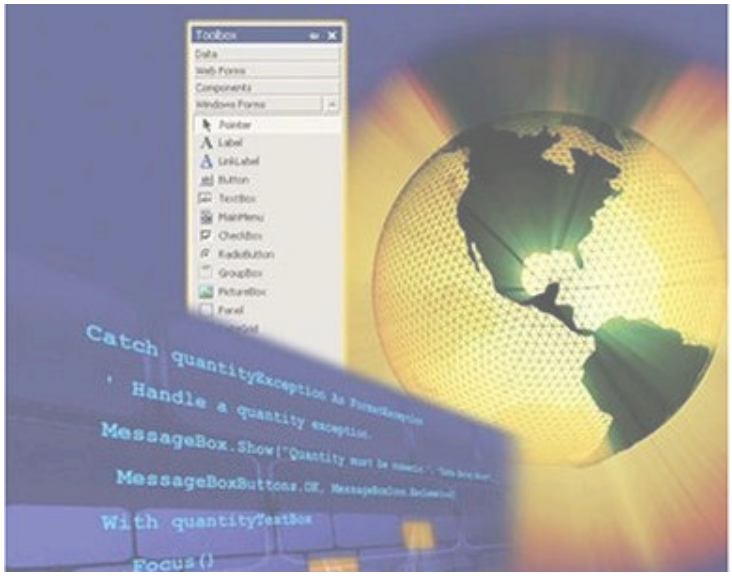


Flowchart

A flowchart is a step-by-step activity on how to solve a problem which consists of a special geometric symbols connected by arrows. Each symbol shows the progression in which the steps take place. Flowchart should “flow” from the top of the page to the bottom. Read Table 1 for descriptions for each symbol.

Symbol	Name	Description
	Terminal	Used to represent the beginning (Start) or the end (End) of a task.
	Input or Output	Used for input and output operations, such as reading and printing. The data to be read or printed are described inside.
	Decision	Used for any logic or comparison operations. Unlike the input/output and processing symbols, which have one entry and one exit flow line, the decision symbol has one entry and two exit paths. The part chosen depends on whether the answer to a question is “yes” or “no”.
	Process	Used for arithmetic and data-manipulation operation. The instructions are listed inside the symbol.
	Connector	Used to join different flow lines.
	Flow line	Used to represent symbols and indicate the flow of logic

Table 1: Basic Flowchart Symbols , adapted from An Introduction to Programming Using Visual Basic 6.0 (2004)



This sample flow chart is to assist you to develop a program. See **Figure 1**.

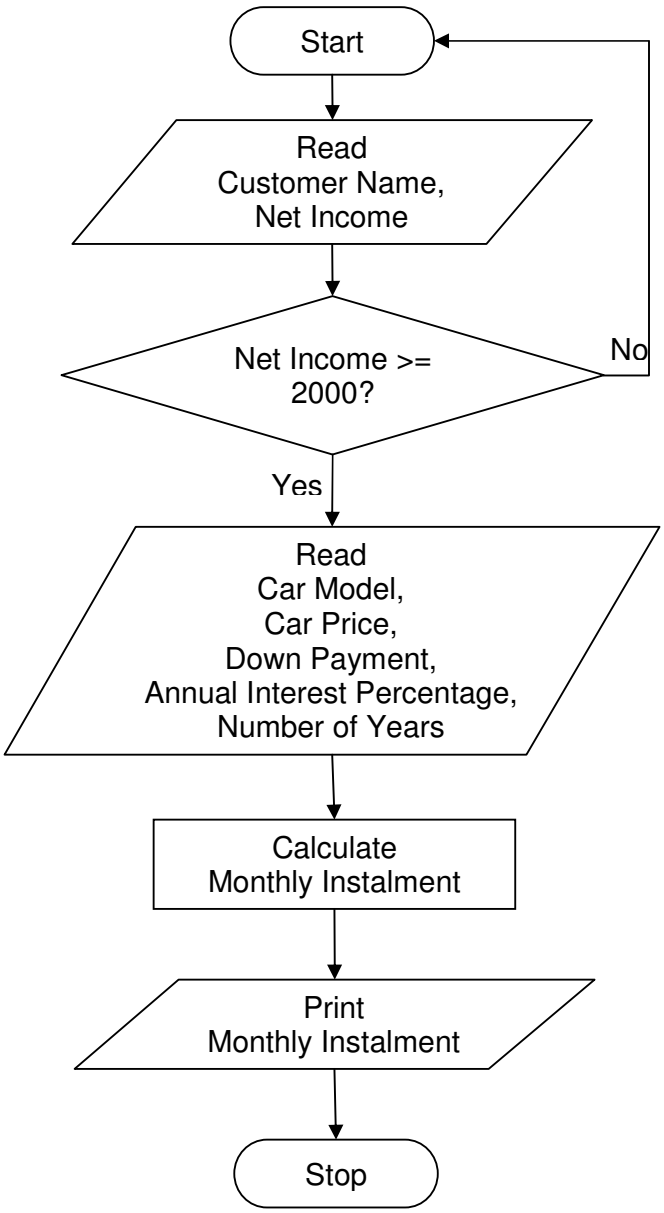
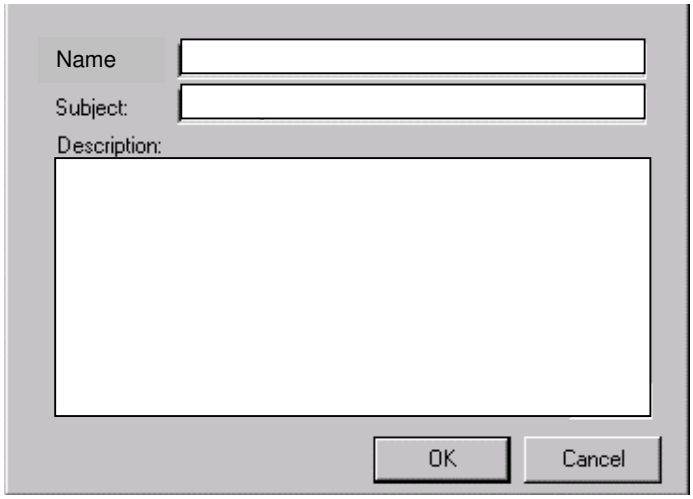


Figure 1: Sample Flowchart

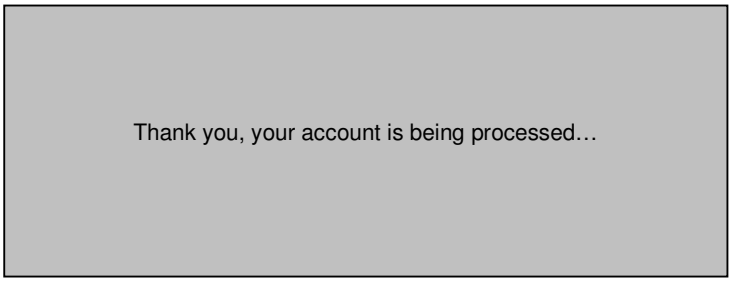
WRITING ACTIVITY

Design User Interface

Design user interface is a simple layout of the screen you want to design for your customer. See the two examples below for input and output.



Design user inteface for Input



Design user interface for Input

Draw your design user interface, based on the input and output you have answered in Problem Analysis Stage (Page 5).

Design user interface for Input

Design user interface for Output

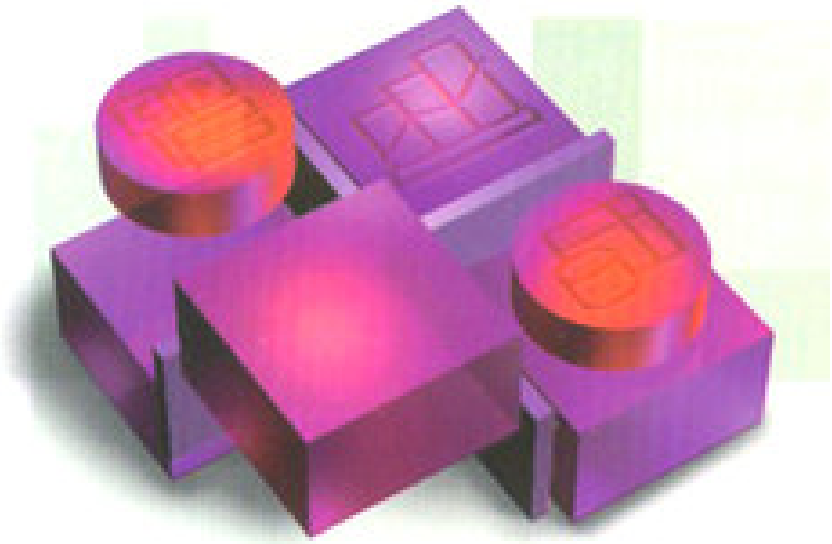
TOPIC : Coding

LEARNING OUTCOMES :

By using this module, you will be able to:

- Create user interface for the new program
- Set the properties for each object in your user interface
- Write program code
- Save and run the program
- Build an executable file

DURATION : 14 periods



HANDS-ON ACTIVITY

What is coding

Coding is the next step after the program logic has been formulated using the appropriate programming language. It also refers to entering the list of commands that becomes a computer program.

Starting a project

In this activity you'll start building Enviro program by creating a new project and then using controls in the toolbox to construct the user interface.

1. After clicking on the Visual Basic program, double click on **Standard EXE** button as shown in **Figure 2**.

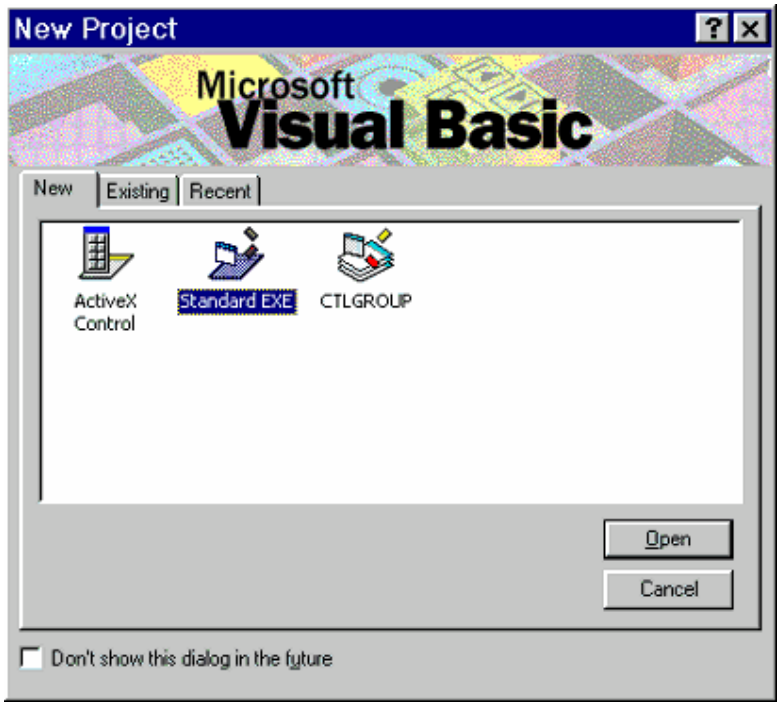


Figure 2: Standard EXE screen

2. After that you will see the screen shown in **Figure 3**.

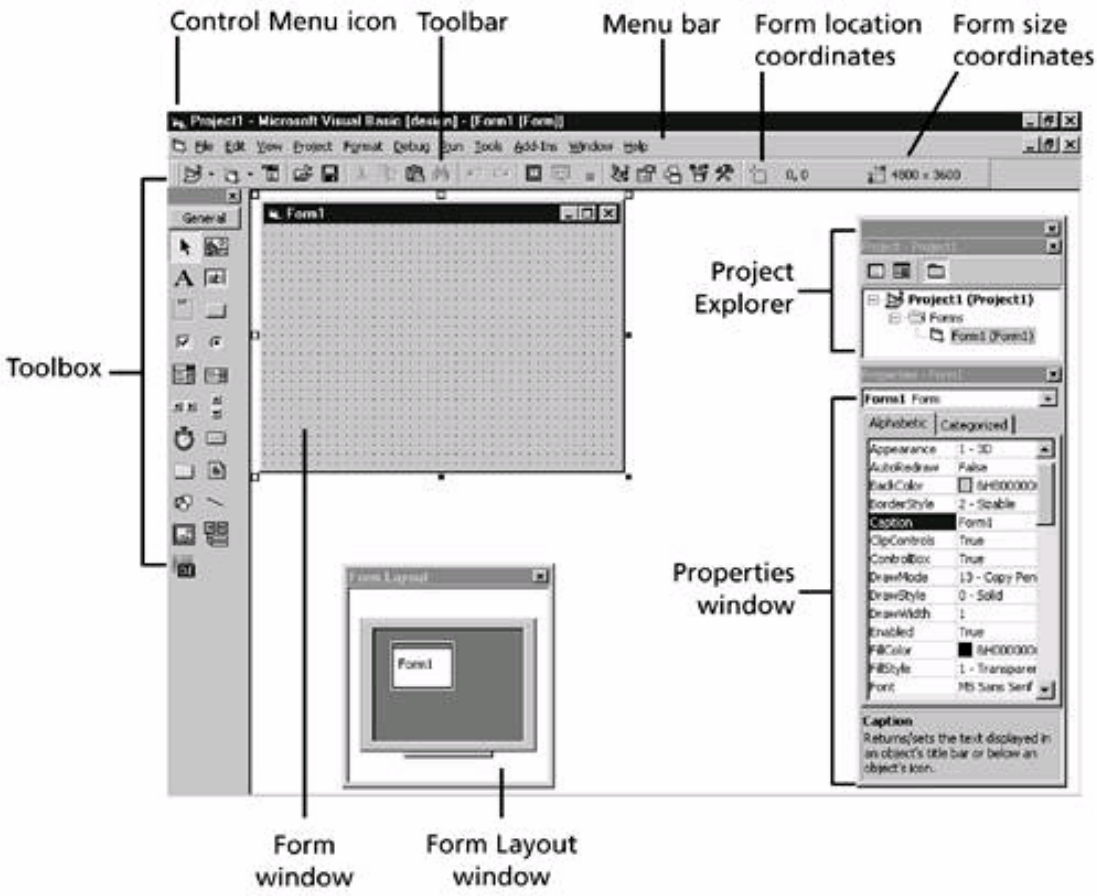


Figure 3: The Visual Basic Interface

3. To start a new programming project, on the File menu, click the New Project command.
4. Click OK to create a standard 32-bit Visual Basic application.

5. The picture shown in **Figure 4** is about the most useful toolbox control you will use in your Visual Basic program:

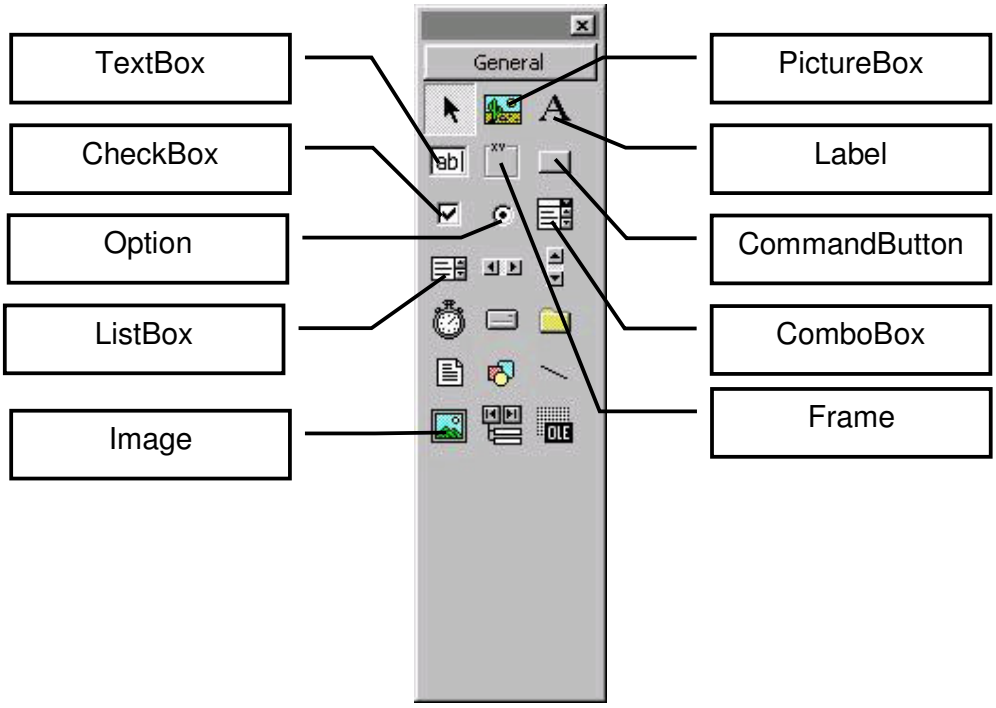


Figure 4: The most useful Visual Basic controls in the Toolbox.

Adding command buttons

1. To add a command button object on the form, click the CommandButton control in the toolbox, then drag the button and place it on the form.
2. You will see the screen which is shown in **Figure 5**, after you have added the CommandButton.

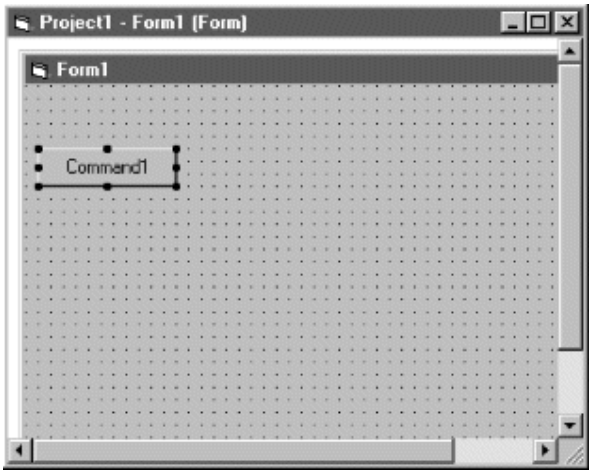


Figure 5: Your first command button

3. To change the settings for the command button, click the first command button (Command1) on the form.
4. Double-click it to open the Properties window title bar.
5. The Properties window is enlarged to full size, as shown in the illustration on **Figure 6**.

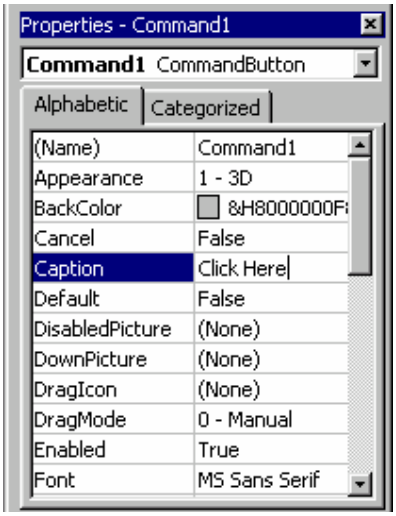


Figure 6: Command1 Properties window

6. The Properties window lists the settings for the first command button. These include settings for the background color, caption, font height and width of the command button.
7. Double-click the Caption property in the left column of the Properties window.
8. The current Caption setting ("Command1") is highlighted in the Properties window.
9. Type **Your Instalment** and press Enter.
10. The Caption property changes to " Your Instalment " in the Properties window and on the form.
11. Add another two new command buttons by repeating step 1.
12. Change their properties by repeating step 4 – 10, according to the table shown in **Table 2**.

Control Tool	**Property**	Object
CommandButton2	Name Caption BackColor	cmdReset Reset &H80000000F&
CommandButton3	Name Caption BackColor	cmdExit Exit &H80000000F&

Table 2: CommandButton2 and CommandButton3 Properties details

13. When you have finished adding all the command buttons, you will see the result as shown in **Figure 7**.

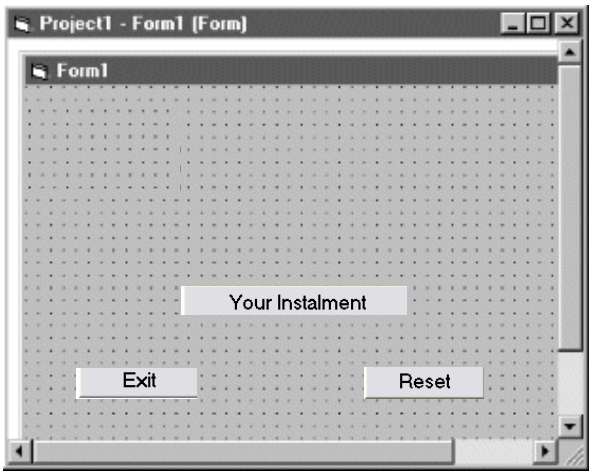


Figure 7: Three finished command buttons

> You can delete an object by selecting the object on the form and then pressing Del key.

Adding labels

A label is a special user interface element designed to display text, numbers, or symbols when a program runs.

1. Click the Label control in the toolbox, and then drag it onto the form, like the shown in **Figure 8**.

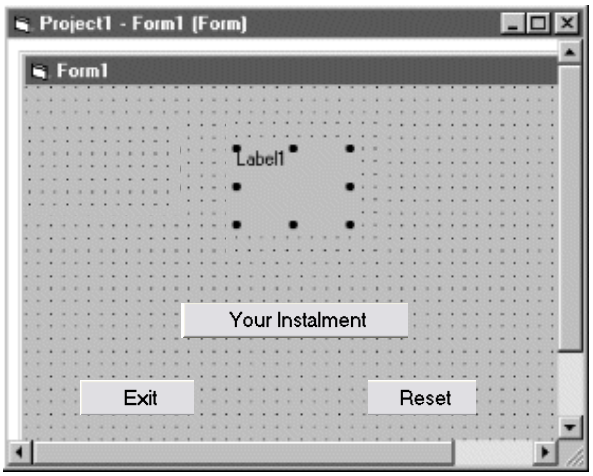


Figure 8: Your first Label

2. Change the properties for the labels in the program. Click the first number label,
3. Click the **Name** property, and type **lblTitle** into the property**.**
4. Double-click the **Caption** property, and modify it into **Enviro Car Enterprise**.
5. Click the **BackColor** property and change it into **&H00FFFF00&**. See the result on **Figure 9**

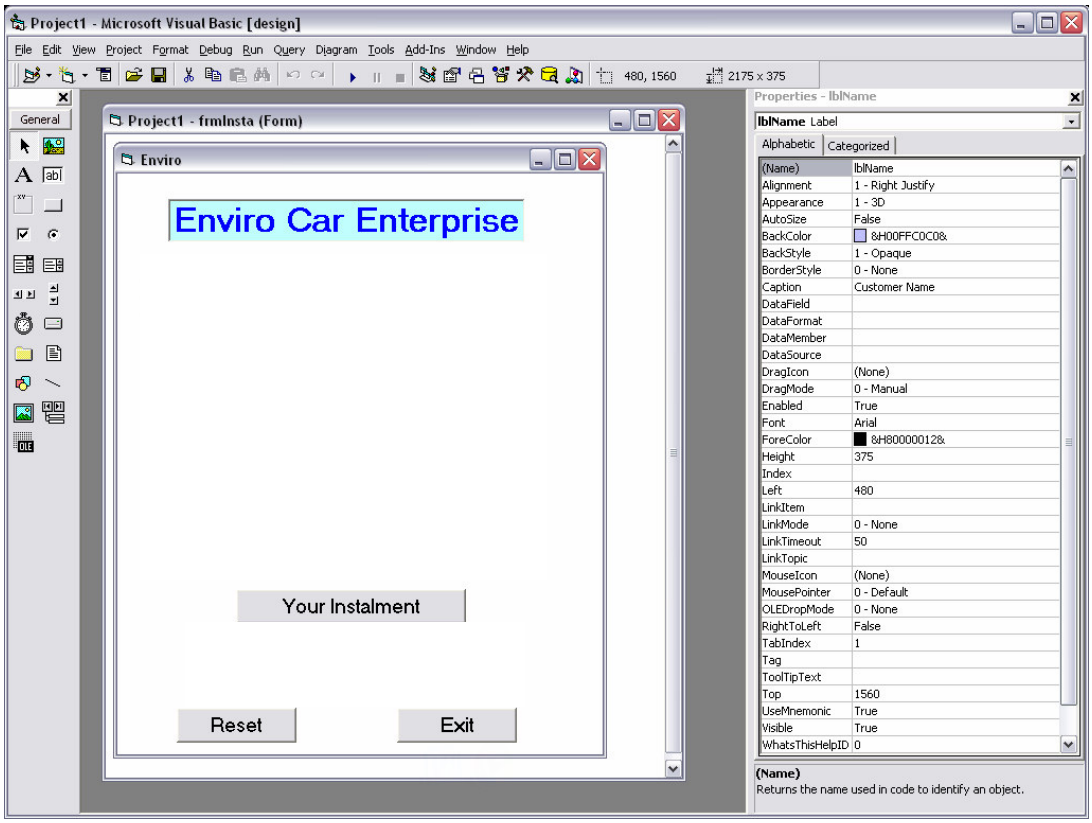


Figure 9: The newly changed label title

6. You have finished with the first label. Now you will have to change the Name, Caption and BackColor properties of the last nine labels. Create nine more labels, named Label2, Label3, Label4, Label5, Label6, Label7, Label8, Label9 and Label10 on the form by repeating step 1.

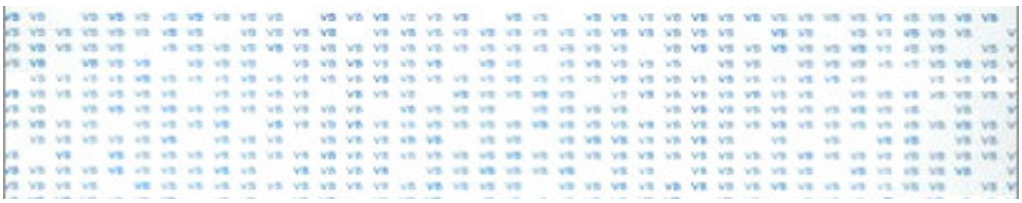
7. Change their properties by repeating step 2 – 5, according to the table shown in **Table 3**.

Control Tool	Property	Object
Label2	Name Caption BackColor	**lblName** **Customer name** **&H00FFFF00&**
Label3	Name Caption BackColor	**lblIncome** **Monthly Net Income** **&H00FFFF00&**
Label4	Name Caption BackColor	**lblModel** **Car Model** **&H00FFFF00&**
Label5	Name Caption BackColor	**lblPrice** **Car Price** **&H00FFFF00&**
Label6	Name Caption BackColor	**lblPayment** **Down Payment** **&H00FFFF00&**
Label7	Name Caption BackColor	**lblInterest** **Annual Interest** **&H00FFFF00&**
Label8	Name Caption BackColor	**lblGoverment** **Government** **&H00FFFF00&**
Label9	Name Caption BackColor	**lblPrivate** **Private** **&H00FFFF00&**
Label10	Name Caption BackColor	**lblDisplay** **(leave it blank)** **&H00FFFF00&**

Table 3: Label2 – Label10 Properties details

To change the **BackColor** properties for last nine labels, you can change their properties in one step.

1. Hold down the Shift key, and click the second, third to the ninth labels.
2. You will see a set of selection rectangles appearing around each label that you click.
3. When you have selected them all, release the Shift key.
4. Double-click the **BackColor** property, and then type **&H00FFFF00&**.



8. When you have finished, your ten labels should look like those in the following illustration (**Figure 10**).

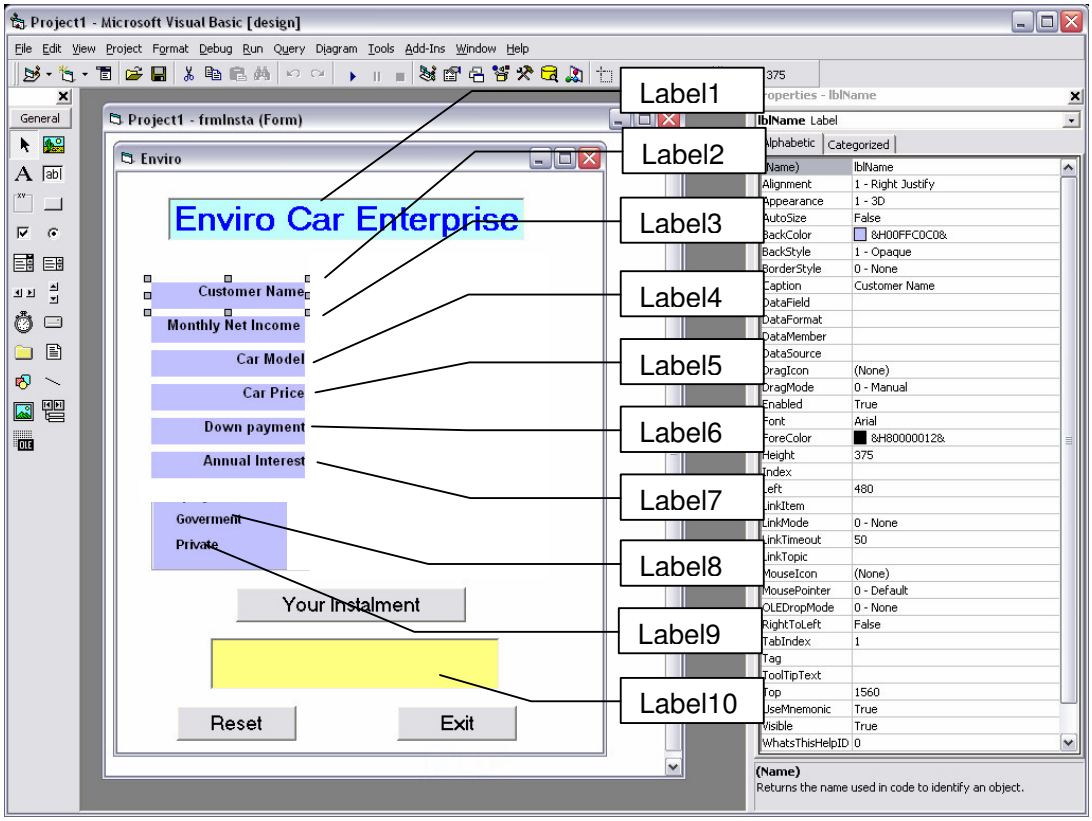
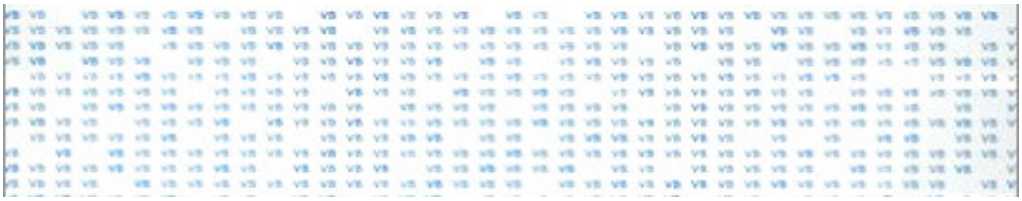


Figure 10: Finished labels



Adding textbox

A textbox is a special user interface element designed to receive input from the user as well as to display the output. It can handle text and numbers but not images.

1. Click the TextBox control in the toolbox, and then drag it onto the form, like the one shown in **Figure 11**.

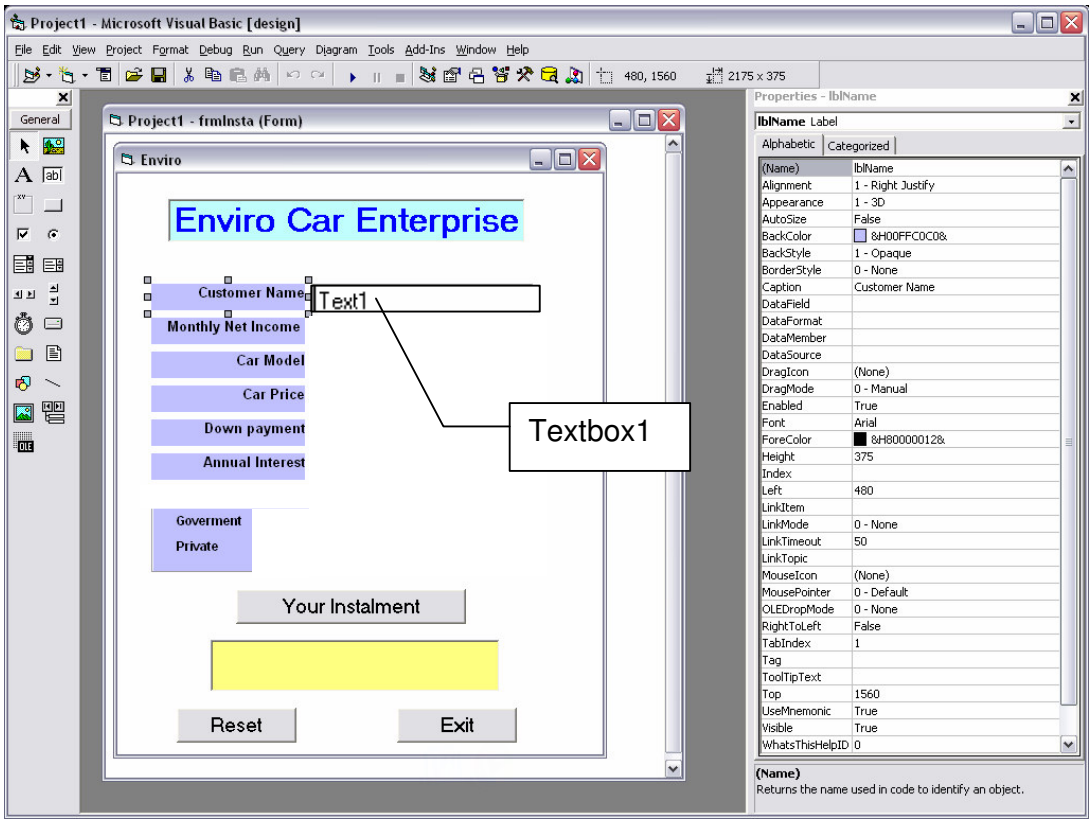


Figure 11: Your first TextBox

2. Change the properties for the textbox in the program.
3. Click the **Name** property, and type **txtName** into the property**.**
4. Click the **BackColor** property and change it into **&H80000005&**.
5. You have finished with the first label. Now you need to change the Name and BackColor properties of the last four textboxes. Create four more textboxes, named TextBox2, TextBox3, TextBox4 and TextBox5 on the form by repeating step 1.

6. Change their properties by repeating step 3 and step 4, according to the table shown in **Table 4**.

Control Tool	Property	Object
TextBox2	Name BackColor	**txtModel** **&H80000005&**
TextBox3	Name BackColor	**txtPrice** **&H80000005&**
TextBox4	Name BackColor	**txtPayment** **&H80000005&**
TextBox5	Name BackColor	**txtInterest** **&H80000005&**

Table 4: TextBox2 – TextBox5 Properties details

7. When you have finished, your five text boxes should look like those in the following illustration (**Figure 12**).

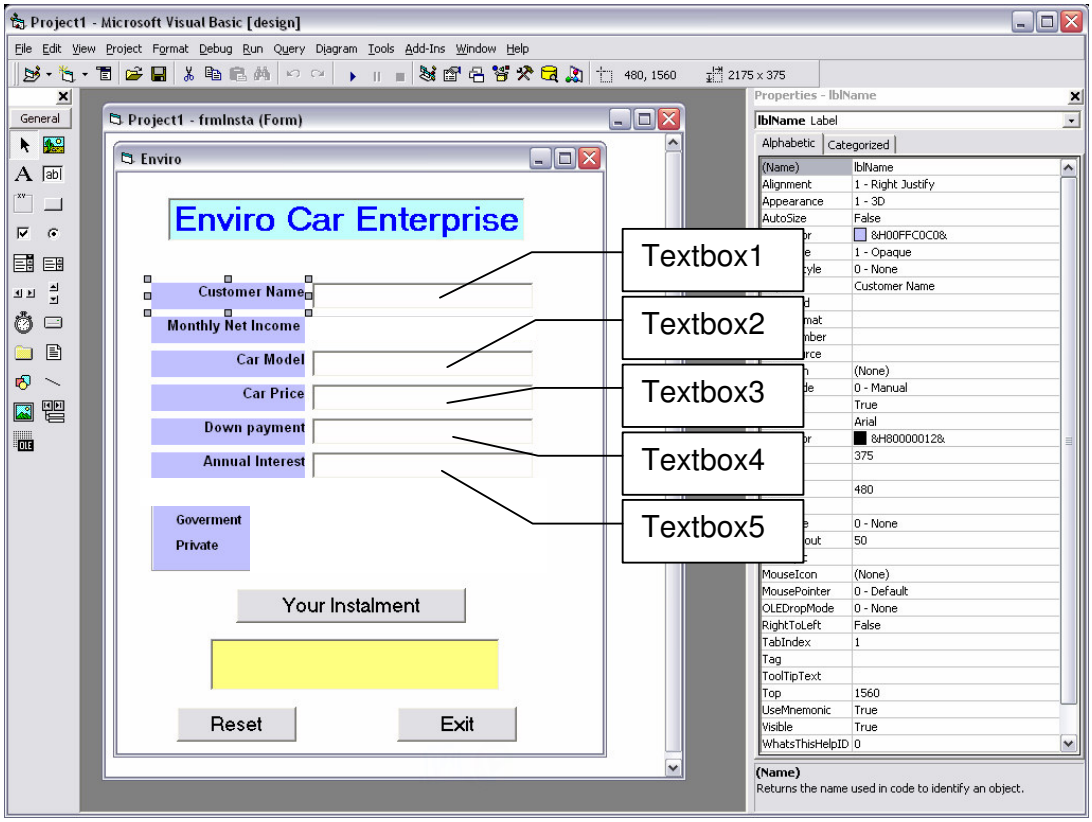


Figure 12: Finished textboxes

Adding Combo Box

A Combo Box is a special user interface element designed to present a list of items where the user can click and select the items from the list. However, the user needs to click on the small arrowhead on the right of the combo box to see the items which are presented in a drop-down list.

1. Click the **ComboBox** control in the toolbox, and then drag it onto the form, like the one shown in **Figure 13**.

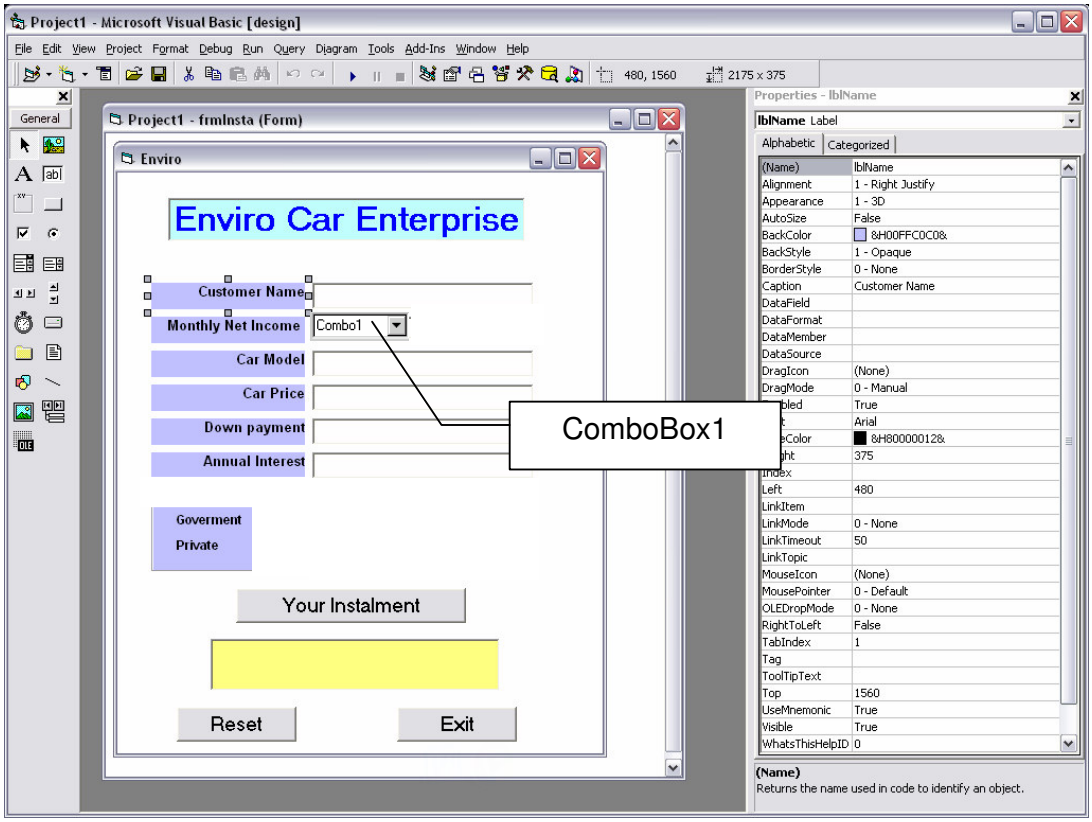
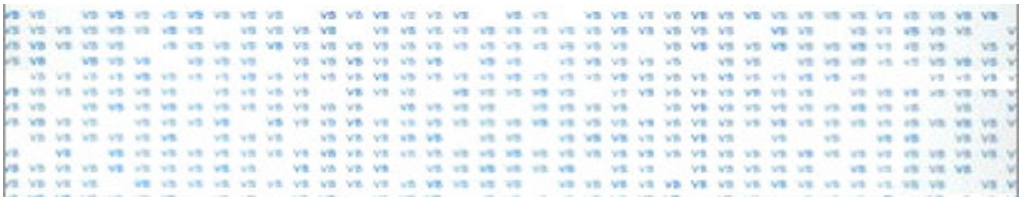


Figure 13: Your first ComboBox

2. Change the properties for the combo box in the program.
3. Click the **Name** property, and type **cboIncome** into the property**.**
4. Click the **BackColor** property and change it into **&H80000005&**.



5. When you have finished, your combo box should look like the following illustration (**Figure 14**).

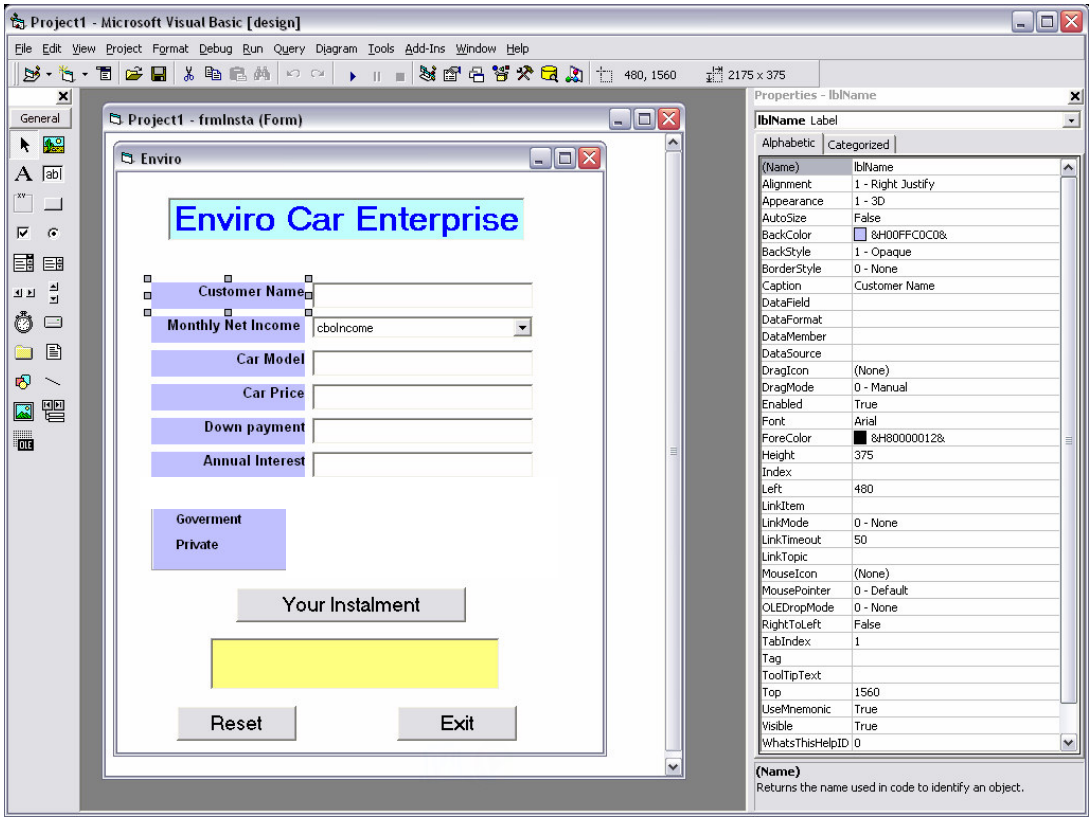
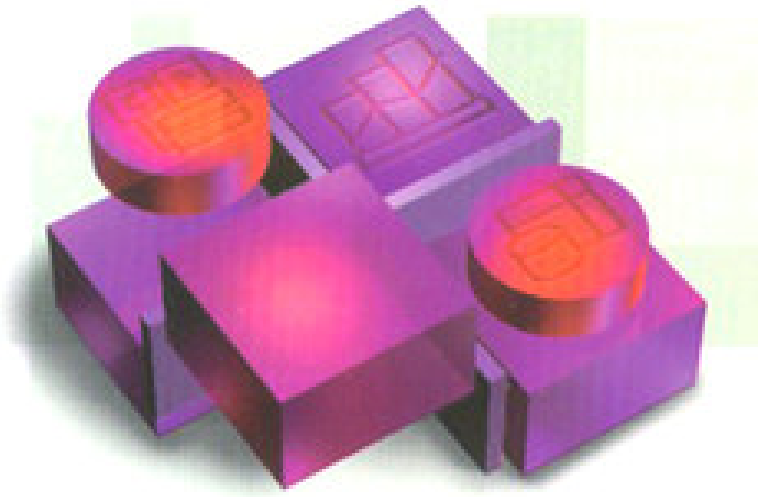


Figure 14: Finished Combo Box



Adding Option

An Option is a special user interface element designed to let the user select it as one of the choices.

1. Click the **Option** control in the toolbox, and then drag it onto the form, like the one shown in **Figure 15**.

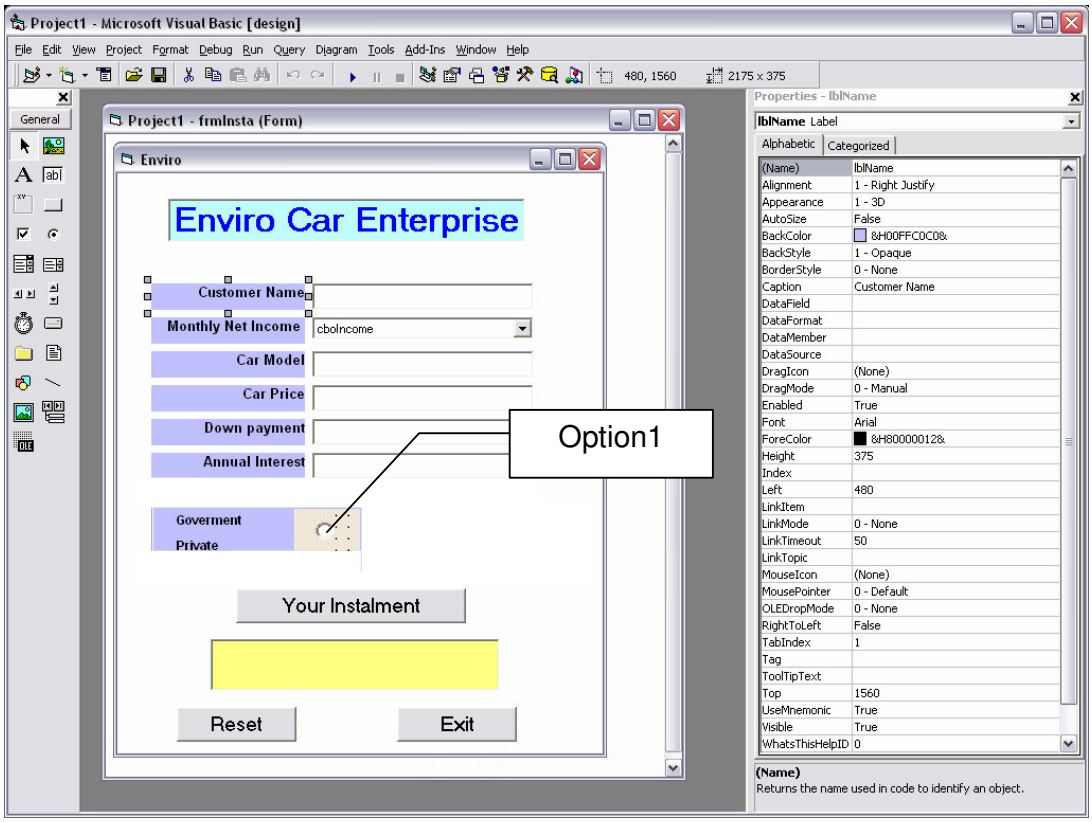


Figure 15: Your first ComboBox

2. Change the properties for the option in the program.
3. Click the **Name** property, and type **optG10** into the property**.**
4. Click the **Caption** property, and type **10years** into the property**.**
5. Click the **BackColor** property and change it into **&H00FFC0C0&**.
6. You have finished with the first option. Now you need to change the Name, Caption and BackColor properties of the last four options. Create four more options, named Option2, Option3, Option4 and Option5 on the form by repeating step 1.

7. Change their properties by repeating step 3 – 5, according to the table shown in **Table 5**.

Control Tool	Property	Object
Option2	Name Caption BackColor	**optG7** **7years** **&H00FFC0C0&**
Option3	Name Caption BackColor	**optG5** **5years** **&H00FFC0C0&**
Option4	Name Caption BackColor	**optP7** **7years** **&H00FFC0C0&**
Option5	Name Caption BackColor	**optP5** **5years** **&H00FFC0C0&**

Table 5: Option2 – Option5 Properties details

8. When you have finished, your options should look like the following illustration (**Figure 16**).

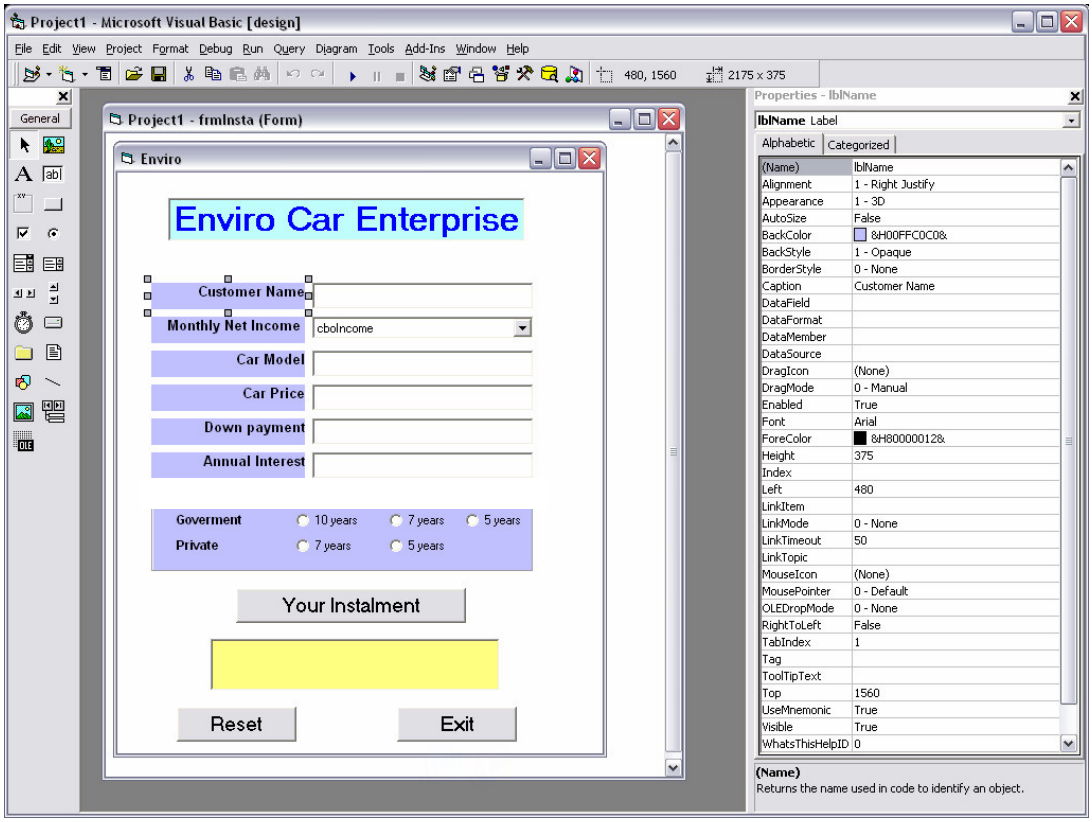


Figure 16: Finished Options

Adding Frame

A Frame is a special user interface element designed to draw a border around objects.

1. Click the **Frame** control in the toolbox, and then drag it onto the form, like the one shown in **Figure 17**.

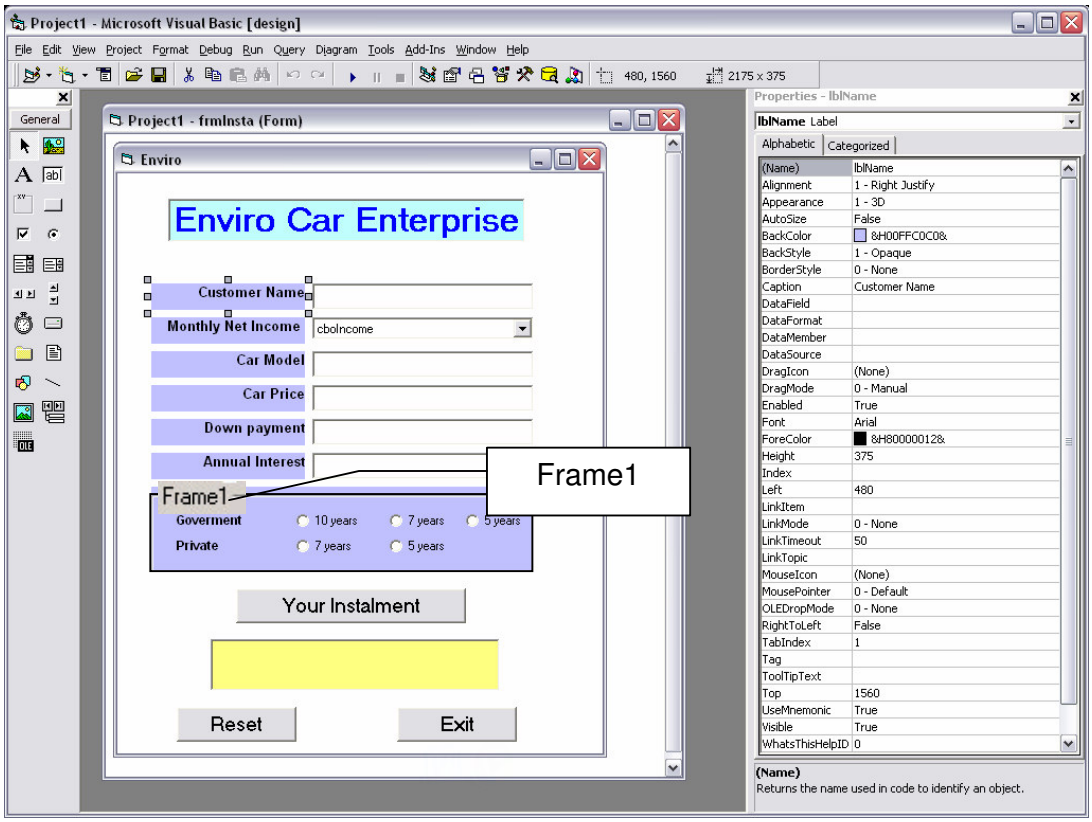
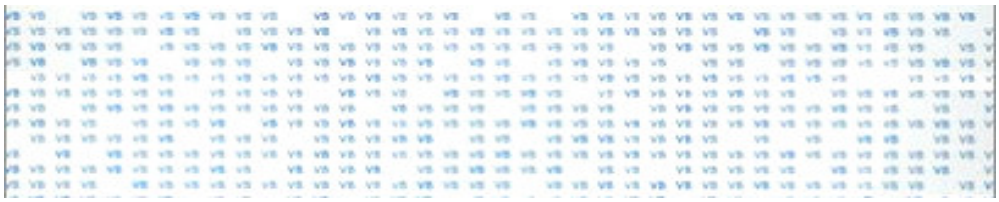


Figure 17: Your first Frame

2. Change the properties for the frame in the program.
3. Click the **Name** property, and type **fraEmploy** into the property**.**
4. Click the **Caption** property, and type **Employee of** into the property**.**
5. Click the **BackColor** property and change it into **&H00FFC0C0&**.



6. When you have finished, your options should look like the following illustration (**Figure 18**).

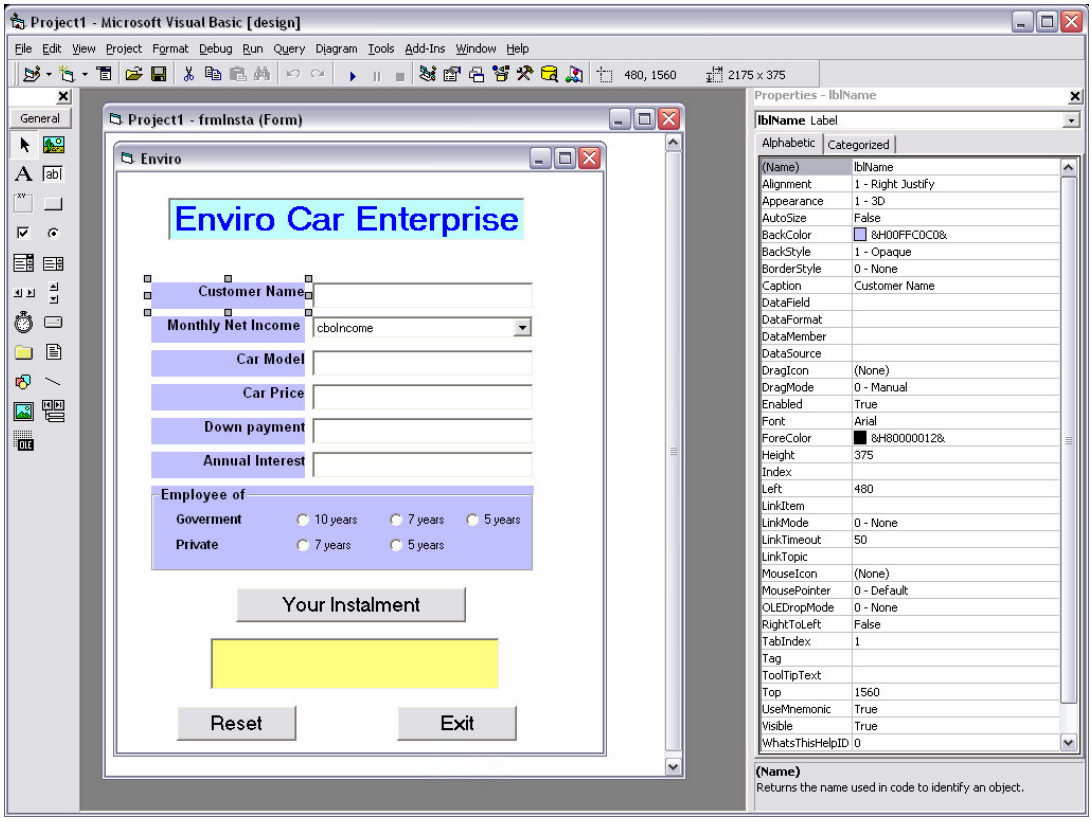


Figure 18: Finished Frame

Writing the code

Now you are ready to write the code for the Enviro program. Your objects (command buttons and labels) are ready to receive input from the user and process it automatically. However, you will need the codes that actually calculate the installment payment and displays it in a box.

The **Code window** is a special window in the programming environment that you use to enter and edit Visual Basic program statements. Inside the Code window are program statements that mark the beginning and the end of this particular Visual Basic subroutine, or *event procedure,* a block of code associated with a particular object in the interface.

In the following steps, you will need to enter the program code for Enviro in the Code window.

1. Double click the spaces on the form.
2. Type the details below between **Private Sub Form_Load()** and **End Sub**.

Type this part

```
Private Sub Form_Load()
        optG10.Value = False
        optG7.Value = False
        optG5.Value = False
        optP7.Value = False
        optP5.Value = False
End Sub
```

3. Close this code window to return to your form.
4. Double-click the **Exit command button** on the form.
5. Type **End,** in between **Private Sub cmdExit_Click(Index As Integer)** and **End Sub** and press the Down arrow key.

Type this part

```
Private Sub cmdExit_Click(Index As Integer)
   End
End Sub
```

6. Close this code window to return to your form.
7. Repeat step 4 for **Reset command button**.
8. Type the details below between **Private Sub cmdReset_Click(Index As Integer)** and **End Sub**.

Type this part

```
Private Sub cmdReset_Click(Index As Integer)
   txtName.Text = ""
   cboIncome.ListIndex = -1
   txtModel.Text = ""
   txtPrice.Text = ""
   txtInterest.Text = ""
   txtPayment.Text = ""
   lblDisplay.Caption = ""
   txtName.SetFocus
End Sub
```

9. Close this code window to return to your form.

10. Repeat step 4 for **Your Installment command button**.
11. Type the details below between **Private Sub cmdInstalment_Click()** and **End Sub**.

Type this part

```
Private Sub cmdInstalment_Click()
   Dim Loan As Currency
   Dim Price As Currency
   Dim Interest As Single
   Dim Instalment As Currency
   Dim Payment As Currency

   Price = Val(txtPrice.Text)
   Interest = Val(txtInterest.Text)
   Payment = Val(txtPayment.Text)

If txtName.Text = "" Then MsgBox "Plese enter customer name", vbCritical,
"Name"
      If cboIncome.ListIndex < 1 Then
      lblDisplay.Caption = "Sorry, your income is not elligible to apply a loan"
   Else
      If optG10.Value Then
         Year = 10

      Loan = Price - Payment
      Instalment = (((Loan * (Interest / 100)) * Year + Loan) / (Year * 12))
      lblDisplay.Caption = "RM" & Format(Instalment, "#####0.00")

      ElseIf optG7.Value Or optP7.Value Then
         Year = 7
      Loan = Price - Payment
      Instalment = (((Loan * (Interest / 100)) * Year + Loan) / (Year * 12))
      lblDisplay.Caption = "RM" & Format(Instalment, "#####0.00")


      ElseIf optG5.Value Or optP5.Value Then
         Year = 5
      Loan = Price - Payment
      Instalment = (((Loan * (Interest / 100)) * Year + Loan) / (Year * 12))
      lblDisplay.Caption = "RM" & Format(Instalment, "#####0.00")

      Else
      MsgBox "Please choose your period of year", vbCritical, "Number of
year"

   End If
   End If
End Sub
```

12. Close this code window to return to your form.

Saving the Program

Now that you have completed the Enviro program, you should save it to disk.

To save a program in Visual Basic, click Save Project As on the File menu or click the Save Project button on the toolbar.

1. On the File menu, click the Save Project As command. (You can save your program at any time during the programming process.)
2. The Save File As dialog box appears, prompting you for the name and storage location for your form.
3. Type **Enviro** in the File Name text box, and press Enter.
4. The form is saved under the name **Enviro.frm**. The Save Project As dialog box then appears in **Figure 19**.

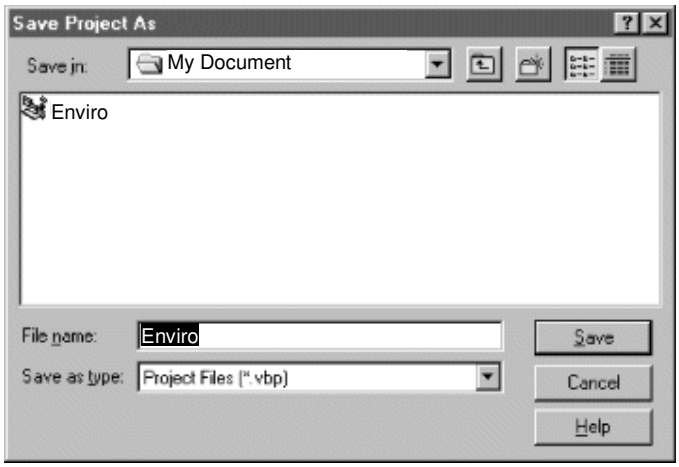
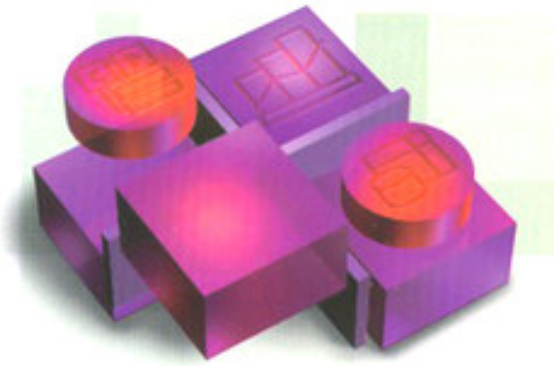


Figure 19: The Save Project As dialog box

5. Type **Enviro** and press Enter.
6. The Enviro project is saved under the name **Enviro.vbp**.



Run the program

1. To run a Visual Basic program from the programming environment, you can click Start on the Run menu, click the Start button on the toolbar, or press F5. The Enviro program runs in the programming environment. The user interface appears, just as you designed it.
2. If Visual Basic displays an error message like the picture shown below, you may still have a typing mistake or two in your program code. See **Figure 20**.

Figure 20: The error message displayed while running a program

3. Try to fix it by comparing the printed version in this module with the one you typed.

Create an executable file

1. On the **File** menu, click the **Make Enviro.exe** command. (Visual Basic adds your program's name to the command automatically.) The Make Project dialog box appears, as in **Figure 21**.

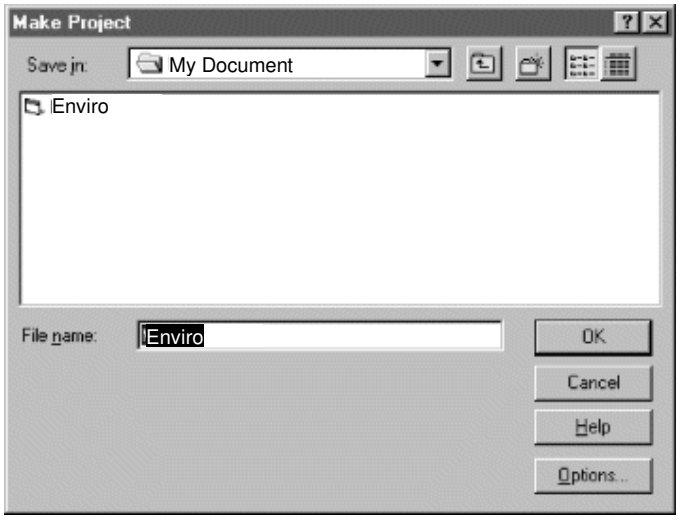


Figure 21: The Make Project dialog box

2. The executable Enviro file is saved under the name **Enviro.exe**.

TOPIC : Documentation

LEARNING OUTCOMES :

By using this module, you will be able to;

1. Create internal documentation
2. Gather all the documents from all the phases into a folio for external documentation

Student's previous knowledge

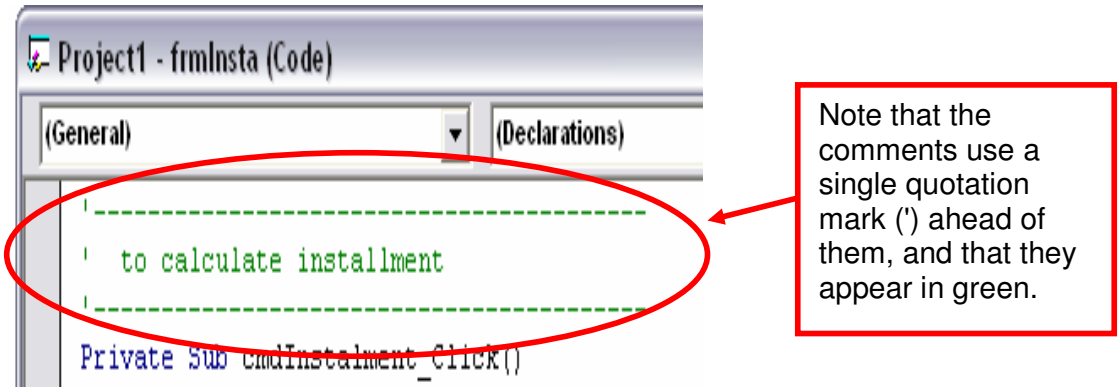
- Student's should able to print screen

Duration: 4 periods

READING ACTIVITY

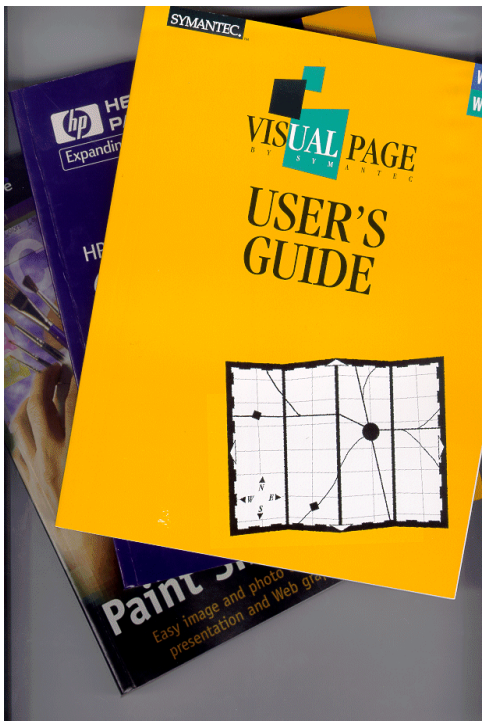
Internal Documentation

Internal documentation is the comment created within the program by the programmer. Internal documentation is used to provide information about the program. Generally, this type of documentation is called a comment, and explains the purpose of a particular line of code or why the programmer used a particular procedure. An example of internal documentation is shown below:



Internal documentation is important because it provides information to programmers who may not be familiar with a particular program.

External Documentation



External documentation is usually a printed manual about the program, how it works, how to install it, or how to use it (user documentation). Sometimes external documentation is provided as a word processing document on the program disk, or as a manual for a user to read. A readme file is a type of external documentation. User manuals and installation manuals are also types of printed external documentation:

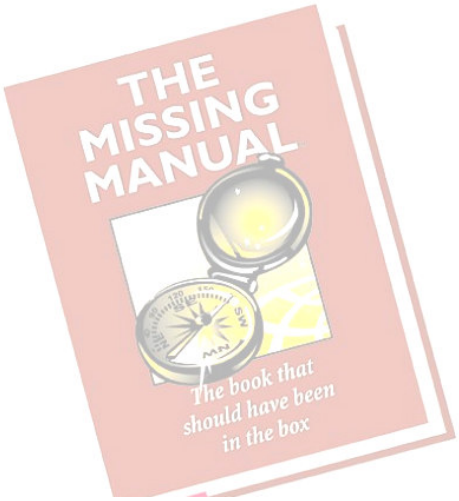
You will be creating both internal and external documentation for your programs as you work through the projects for this module.

WRITING ACTIVITY

Type the minimum requirements for your customer to install in your program. You may add more specifications in the table so that your customer can run your program smoothly.

No.	Specifications	Minimum Requirements
1.	Processor type	
2.	Hard Disk capacity	
3	RAM	
4	Screen Resolution	
5	Operating System type	

You are going to write a simple user guide here. Write the steps on teaching your customer to install the program into his computer.



CONGRATULATIONS !!

YOU HAVE SUCCESSFULLY COMPLETED THIS MODULE. YOU ARE NOW ABLE TO APPLY THE NECESSARY PROGRAMMING SKILL.