



UNESCO-NIGERIA TECHNICAL &  
VOCATIONAL EDUCATION  
REVITALISATION PROJECT-PHASE II



## NATIONAL DIPLOMA IN COMPUTER TECHNOLOGY



### Computer Programming

COURSE CODE: COM113

YEAR I- SEMESTER I

PRACTICAL

Version 1: December 2008

## Table of Contents

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Week 1: How to load the following programming languages; .....           | 3  |
| Week 2: To develop an algorithm for simple programming problems.....     | 6  |
| Week 3: To develop pseudo code for simple programming problems. ....     | 8  |
| Week 4: To draw flowchart for simple programming problems. ....          | 10 |
| Week 5: To write simple programs using different control structure ..... | 16 |
| Week 6: To write simple programs using different control structure ..... | 18 |
| Week 7: To design a program using top-down technique .....               | 20 |
| Week 8: To design a program using top-down technique .....               | 24 |
| Week 9 To code a simple algorithm using any suitable language.....       | 26 |
| Week 10: To code a HLL and translate it to assembly language.....        | 27 |
| Week 11: To code HHL and translate it to assembly language.....          | 29 |
| Week 12: To create a simple bug in a simple program and correct it.....  | 30 |
| Week 13: To write simple structured program .....                        | 32 |
| Week 14: To write simple structured program .....                        | 34 |
| Week 15: To identify properties, events, objects and class in OOP .....  | 38 |

## WEEK 1

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to view some programming languages in a computer

Week 1: How to load the following programming languages;

BASIC, OO-VB, java, C++, etc.

Generally, to load a programming language in a computer, you have to determine whether to load from;

- a. DOS environment or
- b. Windows environment.

- a. To load programming languages from DOS environment, follow these steps;

To load BASIC from command prompt:

- At C:\ type CD **BASIC**, then press enter key. (To change directory)
- At C:\BASIC> type **QBASIC**, then press enter key. (To load the QBASIC program editor)

#### Note that;

- These procedures are the same for all DOS-based programs
- The directory used above may vary depending on the directory name used for specific programming language or as installed.
- Alternatively, right-click on the **ICON** on the desktop, then from the pop-up menu, select **open** option to load the program or simply double-click the **ICON**.

- b. To load programming languages from Windows environment, follow these steps;

- i. To load OOVB from Windows environment:

- Click **Start** on the status bar
- Select **All Programs** (To display program launcher)
- Select **Visual Studio**

- Select and click on **Visual Basic** (To load OOVb)

ii. To load java application from Windows environment:

- Click **Start** on the status bar
- Select **All Programs** (To display program launcher)
- Select and click on **java** (To load JAVA application)

#### Note

\* Any text editor such as word pad, note pad, etc can be used to enter JAVA programs/codes but the '*file name extension*' should be **dot java** (.java). Java compiler is used to compile the program while java virtual machine (jvm) or java interpreter used to run/execute the compiled codes.

iii. To load C++ program from Windows environment:

- Click **Start** on the status bar
- Select **All Programs** (To display program launcher)
- Select **Visual Studio**
- Select and click on **C++** (To load C++ program)

#### Note that;

- These procedures are the same for all Window-based programs
- Alternatively, right-click on the ICON on the desktop, then from the pop-up menu displayed, select on **open** option to load the program or simply double-click the program ICON.

### STUDENTS' ASSESSMENT

1. Did you encounter difficulties while loading any of the programming languages?

---

2. Briefly explain the nature of problem(s) encountered and how you overcome it.

---

---

---

---

3. Give five examples of window-based programming languages.

- i. \_\_\_\_\_
- ii. \_\_\_\_\_
- iii. \_\_\_\_\_
- iv. \_\_\_\_\_
- v. \_\_\_\_\_

4. From your practical laboratory experience, state the merits and demerits of DOS-based programming languages and Window-based programming languages.

---

---

---

5. In loading programming languages in a computer, which of the methods below

- a. Double-clicking the program ICON on the desktop
- b. Right-click on the ICON, then select **open** option
- c. Click on **Start** on status bar, select **All Programs**, then select the **Program**

**WEEK 2****SPECIFIC LEARNING OUTCOMES****To understand:**

- ✓ How to develop an algorithm for simple programming problems.
- ✓ : To develop an algorithm for simple programming problems.

1. Develop an algorithm to find the average of four numbers stored in variables A, B, C, D.

**How to solve this problem**

- Step 1: Get the *values* for the variables (A, B, C, D)
- Step 2: Compute the *Sum* by adding the numbers (A+B+C+D)
- Step 3: Compute the *average* by dividing the *summed value* by **4** (sum/4)
- Step 4: Display the average (print average)
- Step 5: End Process

2. Write an algorithm to convert the length in feet to centimeter.

**How to solve this problem****i. Algorithm**

- Enter the length in feet (*Lft*)
  - Calculate the length in *Lcm* by multiplying *Lft* with 30
  - Print Length in *cm* (*Lcm*)
  - Stop
3. Students are to develop an algorithm to calculate simple interest (**SI**) on x principal at y% rate and z years.

**Hint:**

$$\text{Simple Interest} = \text{Principal} \times \text{Rate} \times \text{Time}$$

How to write an Algorithm this problem

- Begin the process by initializing the variables (x, y, z)
- Input **enter principal (x)**
- Input **enter rate (y)**
- Input **enter time (z)**
- Compute **Simple Interest (SI)** by multiply **Principal(x)** with **Rate(y) and Time(z)**
- Display the **Simple Interest** (Print **SI**)
- Stop process.

## WEEK 3

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to develop pseudo code for simple programming problems.

#### To develop pseudo code for simple programming problems.

1. Write a pseudo code to find the average of four numbers stored in variables A, B, C, D.

#### How to develop Pseudo code for this problem

- Step 1: Get *values* (A, B, C, D)
- Step 2: *sum*  $\leftarrow A + B + C + D$
- Step 3: *average*  $\leftarrow \text{sum} / 4$
- Step 4: Display *average* (print average)
- Step 5: End Process

2. Students are to develop pseudo code to calculate simple interest (**SI**) on x principal at y% rate and z years.

#### How to develop Pseudo code for this problem

- Step 1: enter x, y, z
- Step 2: **SI**  $\leftarrow x * y * z$
- Step 3: display **SI** (calculated)
- Step 4: end it



3. Develop pseudo code to convert the length in feet to centimeter.

**How to develop Pseudo code for this problem**

- Input the length in feet (Lft)
- Calculate the length in an (LCM) by multiplying LFT with 30
- Print Length in Cm (LCM)

Step 1:     Input Lft

Step 2:     Lcm  $\leftarrow$  Lft \* 30

Step 3:     Print Lcm

Step 4:     Stop

## WEEK 4

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to draw flowchart for simple programming problems.

#### To draw flowchart for simple programming problems.

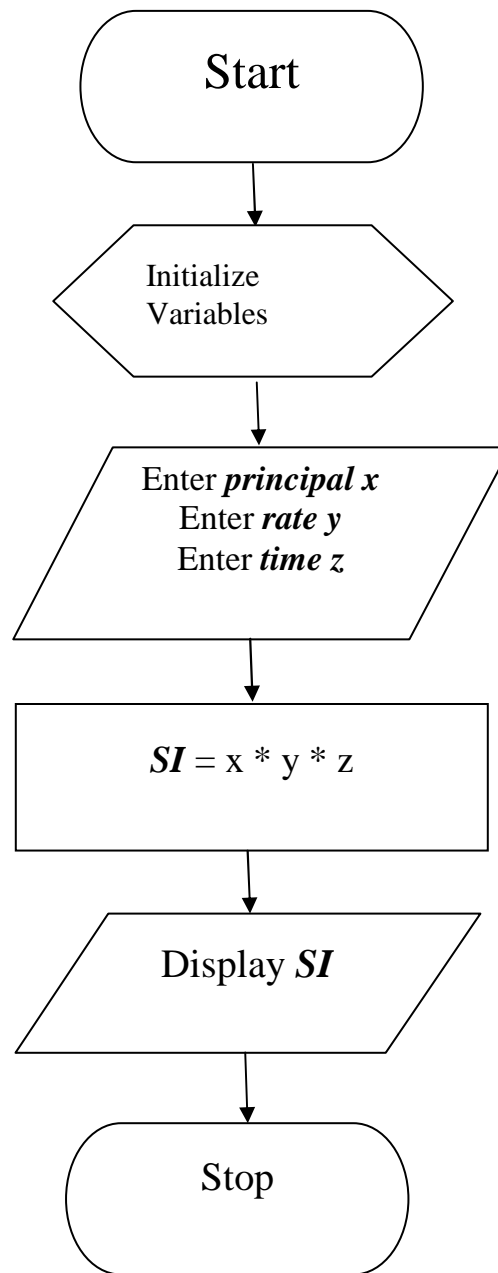
Students should be able to draw flowcharts for simple programming problems.

1. Students are to develop draw flowchart to calculate simple interest (**SI**) on x principal at y% rate and z years.

**Hint:** *Simple Interest = Principal x Rate x Time*

#### How to draw Flowchart

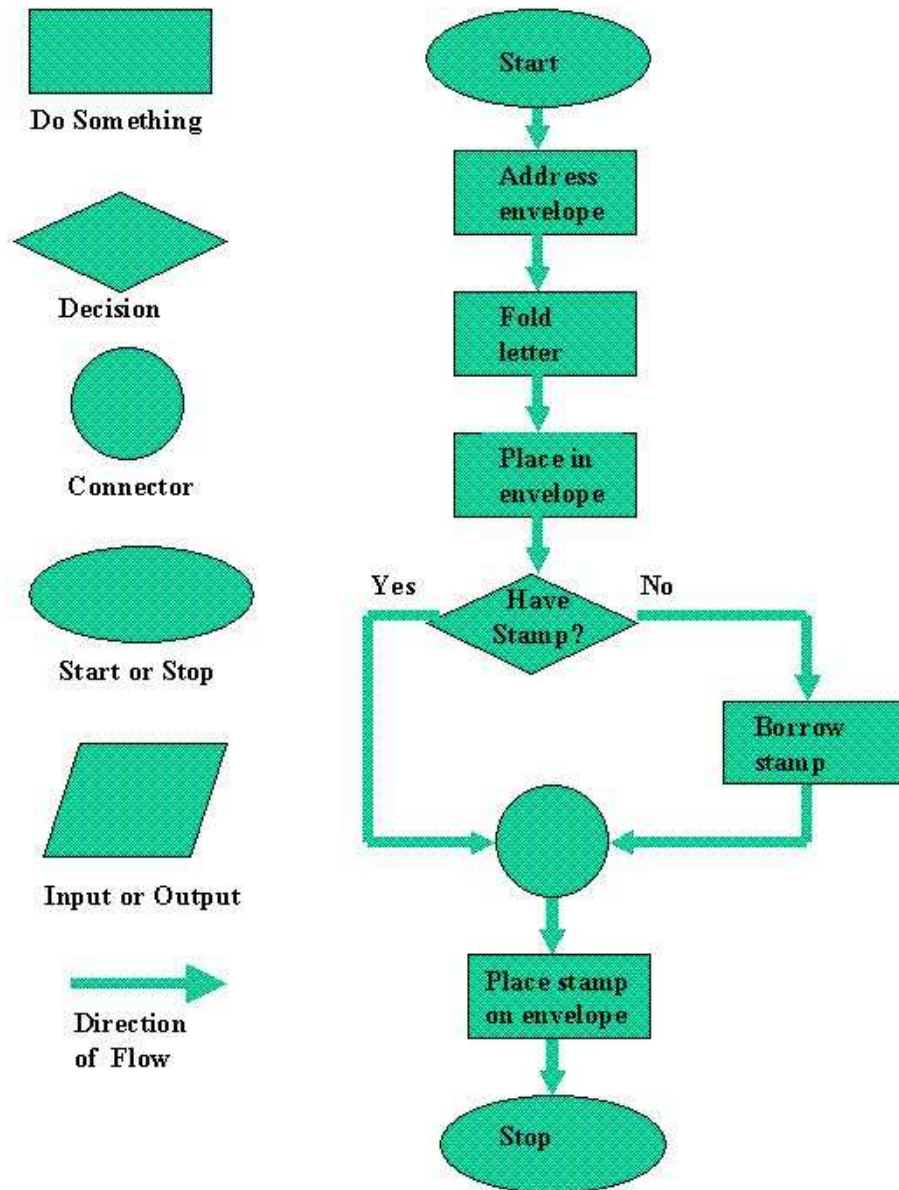
In drawing flowchart, students are to identify the various symbols that are appropriate for each process and each process must be logically connected by arrow symbols.



Student should draw a flowchart for mailing a letter to a friend using appropriate symbols.

## How to draw Flowchart

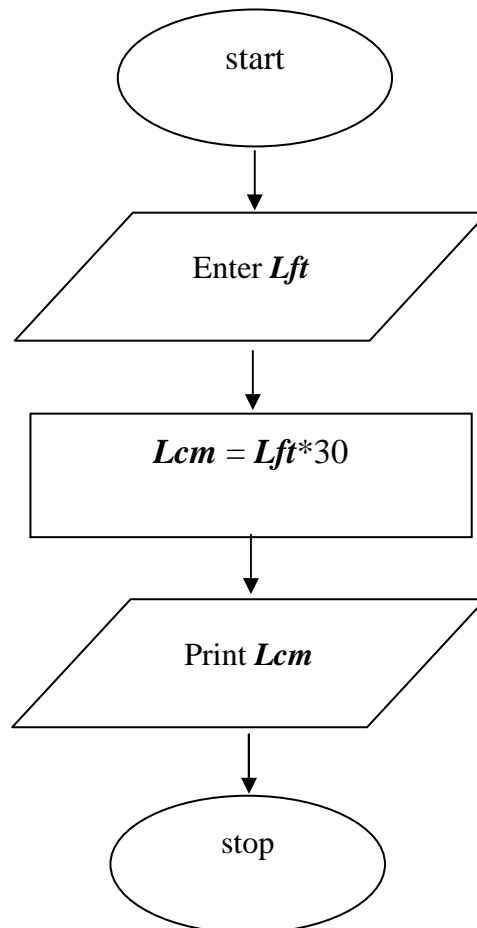
- Identify the each symbol appropriate for a process
- Draw the symbol logically
- Connect the symbols with arrows to indicate the flow of the logic



2. Draw a flowchart to convert the length in feet to centimeter.

## How to draw Flowchart

- Identify the each symbol appropriate for a process
- Draw the symbol logically
- Connect the symbols with arrows to indicate the flow of the logic



**STUDENTS' ASSESSMENT**

**At the end of weeks 2 – 4, the Students should be assessed on the following;**

6. Did you find it difficulties to identify symbol for specific process?

---

---

7. If your answer to question 1 above is *YES*, then state the process.

---

---

---

---

8. State the appropriate symbol used for the following processes/operations?

vi. Assignment: \_\_\_\_\_

vii. Decision: \_\_\_\_\_

viii. Initialization: \_\_\_\_\_

ix. Input and Output: \_\_\_\_\_

x. Terminator: \_\_\_\_\_

xi. Flow of program logic: \_\_\_\_\_

9. From your practical experience, indicate the program design tools that you find more comfortable.

i. Flowchart

ii. Pseudo code

iii. Algorithm

10. Give reason(s) for your answer in question 4 above.

---

---

**WEEK 5****SPECIFIC LEARNING OUTCOMES****To understand:**

- ✓ How to write simple programs using different control structure

To write simple programs using different control structure

1. Write a structured program to determine a student final score in four courses and indicate whether the grade is pass or fail by comparing the average score with 45 (pass mark).

**How to write program in QBASIC Language**

*CLS*

*REM To write a BASIC program to compute average score*

*REM This program is developed on 26 -11 – 2008*

*REM By UNESCO*

*Sum = 0*

*Counter = 0*

*10 INPUT "Enter score:"; score*

*Sum = Sum + score*

*Counter = Counter + 1*

*IF Counter < 4 THEN GOTO 10*

*average = sum/counter*

*IF average >= 45 THEN remark\$ = "Pass" ELSE remark\$ = "Fail"*

*PRINT average*

*PRINT remark\$*

*END*



2. Write a structured program using **for ... next** loop statement to generate integer numbers from 1 to **n**, where **n** could be any number.

```
For counter=1 to n  
  
    display.Text=counter  
  
    counter = counter + 1  
  
Next n
```

3. Write a program using the **For...Next** statement to create a string that contains 10 instances of the numbers 0 through 9, each string separated from the other by a single space. The outer loop uses a loop counter variable that is decremented each time through the loop.

#### How to write this program in VB Language

```
Dim Words, Chars, MyString  
For Words = 10 To 1 Step -1      ' Set up 10 repetitions.  
    For Chars = 0 To 9      ' Set up 10 repetitions.  
        MyString = MyString & Chars      ' Append number to string.  
    Next Chars      ' Increment counter  
    MyString = MyString & " "      ' Append a space.  
Next Words
```

## WEEK 6

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to write simple programs using different control structure

To write simple programs using different control structure

Student are to write simple BASIC program to compare three number;  $a$ ,  $b$ ,  $c$  and determine the least numbers among the 3.

#### How to write this program:

- i. Get Values for variables;  $a$ ,  $b$ ,  $c$
- ii. First: compare two values;  $a$ ,  $b$  to determine the less\_value
- iii. Second: compare the less\_value with  $c$  to determine the least\_value
- iv. Display the least\_value

#### Program

---

```
Cls
REM –This program determine the least of three numbers.
REM – Written by NBTE – Nigeria.
REM – 03 – 12 – 2008
DIM a as integer
DIM b as integer
DIM c as integer
CLS
Input "enter the three values: "; a, b, c
    If a>b then goto 10 else 20
    10 If a<c then goto 30 else 40
    20 if b>c then goto 40 else 50
    30 Print "the less value is: "; a
        Goto 50
    40 Print "the less value is: "; c
        Goto 50
    50 Print "the less value is: "; b
    60 end
```

---

1. Students are to draw flowchart for the program developed for question 1 above.

2. Itemize the problems encountered and state how possible solutions proffered.

---

---

## WEEK 7

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ To be able to design a program using top-down technique

Week 7: To design a program using top-down technique

At the end of week 7, Students should be able to draw flowcharts for simple programming problems and demonstrate;

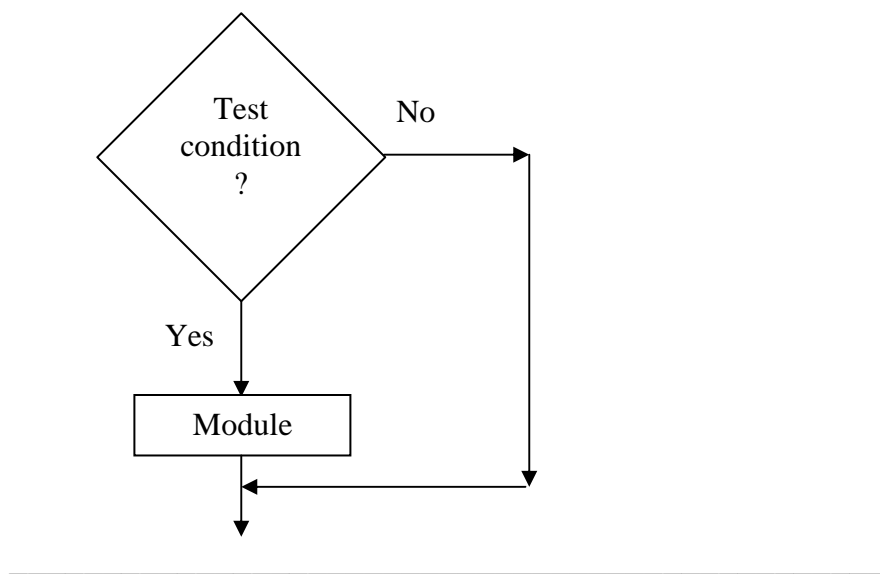
- a. Single alternative
- b. Double alternative structure
- c. Multiple alternatives:

#### (a) Single alternative

This structure has the form:

*[IF condition, Then:]*

*[End of IF structure]*



```

IF (age > 20)
    Lable1.caption = "Qualified"
ENDIF
    
```

---

From the above, if the condition is satisfied, the program displays **Qualified**, otherwise the execution continues from the statement after **ENDIF** statement

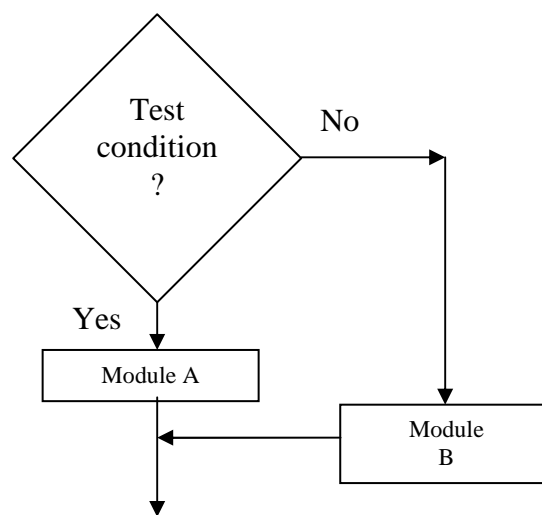
(b) Double alternative: This structure has the form:

```

IF (Condition, then)
    [Module A]

Else
    [Module B]

ENDIF structure]
    
```



---

*IF (age > 20)*

*Lable1.caption = "Qualified"*

*ELSE*

*Lable1.caption = "Unqualified"*

*ENDIF*

---

From the above, if the **IF** condition holds, then module **A** will be executed, otherwise module **B** will be executed. That is whatever condition holds, an action will occur (i.e module A or module B must take place).

(c) Multiple alternatives:

This structure has the form:

*IF Condition (1), then*

*[Module A,]*

*Else if condition (2), then:*

*[module AA]*

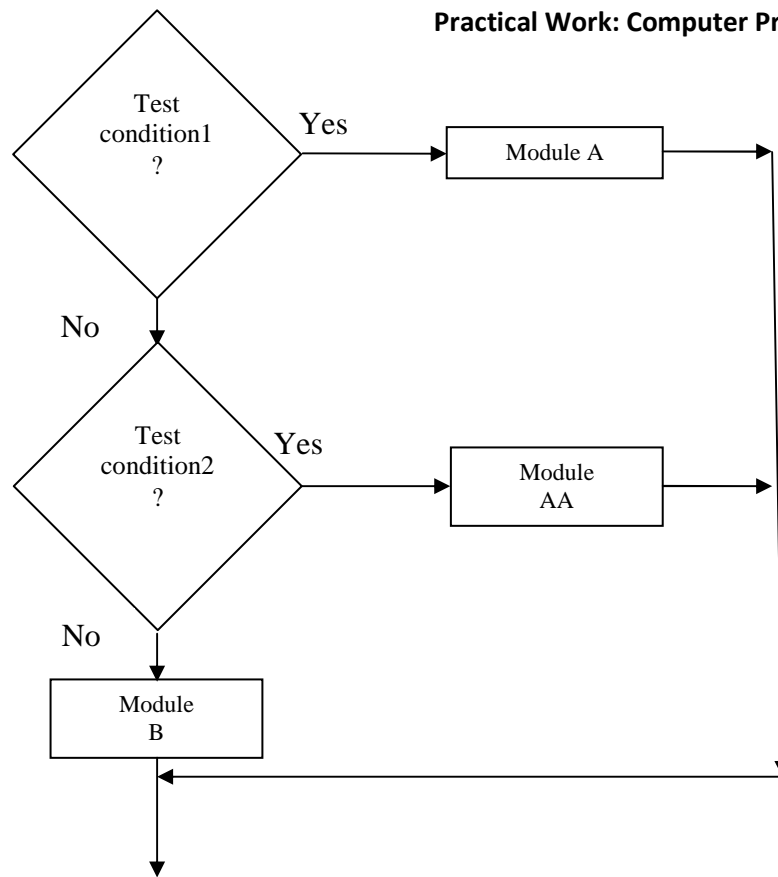
*:*

*Else if condition (m), then;*

*[module Am]*

*Else [module B]*

*(END IF structure)*




---

*IF (age > 20)*

*Lable1.caption = "qualified"*

*ELSEIF (height =6)           'ie 6ft*

*Lable1.caption = "under consideration"*

*ELSE*

*Labell.caption = "unqualified"*

*ENDIF*

---

From the above, if the **IF** condition1 holds, then module **A** will be executed and the process ends. IF condition2 holds, then module **AA** will be executed and the process ends. Otherwise module **B** will be executed.

## WEEK 8

### SPECIFIC LEARNING OUTCOMES

#### To understand:

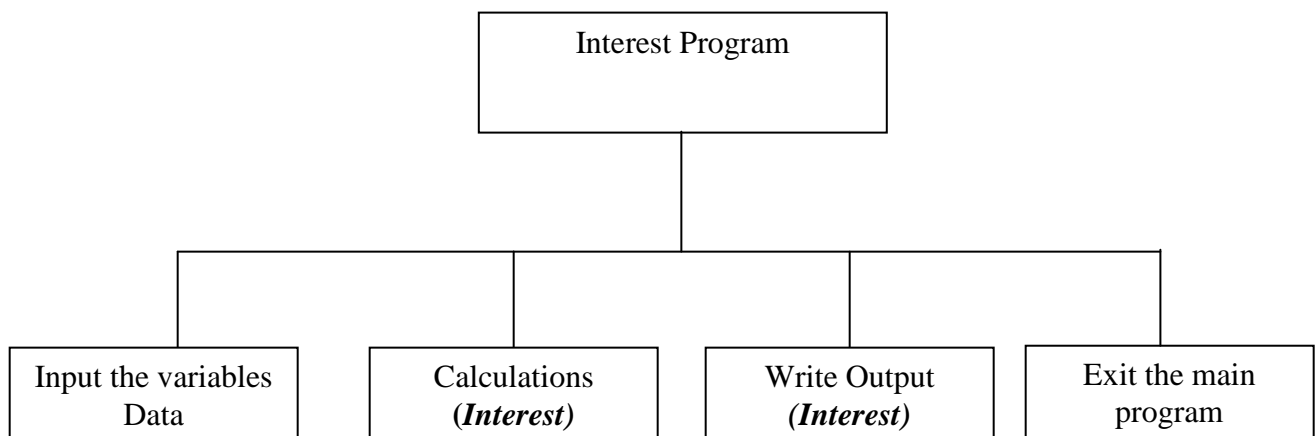
- ✓ To be able to design a program using top-down technique

To design a program using top-down technique

At the end of this week the students should be able to use modular (subroutines) approach of program plan for calculating simple interest on ₦2,000.00 for 3 years at the rate of 7%.

#### How to solve this problem:

- i. Divide the problem into appropriate segments/modules
  - a. Input module to accept all the variables (*values*)
  - b. Process module to manipulate/calculate the *Interest*
  - c. Output module to produce the *Interest*
- ii. Develop each module separately
- iii. Join the modules under a *single controlled program*
- iv. Represent steps [i – iii] in a simple ‘program plan’ diagram as below:





3. Students are to write simple program for each module
4. With the aid of a diagram, illustrate the hierarchical plan of adding and multiplying 4 different numbers; say,  $w$ ,  $x$ ,  $y$ ,  $z$ .
5. Enumerate the advantages of modular programming.

---



---



---



---



---

## **WEEK 9**

### **SPECIFIC LEARNING OUTCOMES**

#### **To understand:**

- ✓ To be able to code a simple algorithm using any suitable language

#### **To code a simple algorithm using any suitable language**

At the end of this week, the students should list and discuss practical experiences gained from the various stages involved in program development.

---

---

---

---

---

## WEEK 10

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to code a very simple high level language and translate it to assembly language.

#### To code a HLL and translate it to assembly language.

1. At the end of this week, students are to write codes in C++, BASIC, FOTRAN, COBOL, PASCAL to add, subtract and multiply two distinct numbers.

#### How to solve this problem

##### i. Code in C++:

$$Z = X + Y;$$

$$Z1 = X - Y;$$

$$Z2 = X * Y;$$

##### ii. Code in BASIC:

$$LET Z = X + Y$$

$$LETE Z1 = X - Y$$

$$LET Z2 = X * Y$$

##### iii. Code in FORTRAN:

$$Z = X + Y$$

$$Z1 = X - Y$$

$$Z2 = X * Y$$

**iv. Code in COBOL**

*ADD Y TO X GIVEN Z*

*SUBTRACT Y FROM X GIVEN Z1*

*MULTIPLY X AND Y GIVEN Z2*

**v. Code in PASCAL**

*Z := X + Y*

*Z1 := X - Y*

*Z2 := X \* Y*

2. Enumerate the levels of computer languages and give example of each level.

---

---

---

---

---

## WEEK 11

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to code a very simple high level language and translate it to assembly language.

To code HHL and translate it to assembly language.

Students should give detail steps of how to load QBASIC and assembly language

1. Write a simple BASIC program to perform addition, multiplication, subtraction and division on two values –  $x$  and  $y$ . where  $x = 8$  and  $y = 2$ .
2. Convert the program written in 1 above to machine language.
3. Convert the program written in 1 above to assembly language.

## WEEK 12

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to create a simple bug in a simple program and correct it

To create a simple bug in a simple program and correct it

1. At the end of this week 12, Students should be able identify syntax errors, logic errors, and run time, etc, in a program.

Identify syntax errors in the program below and state reasons why the program may not run.

#### How to identify error:

- Note that in a nest **For...Next** loops one **For...Next** loop is placed within another.
- Critically examine program line by line to locate where bugs exist

---

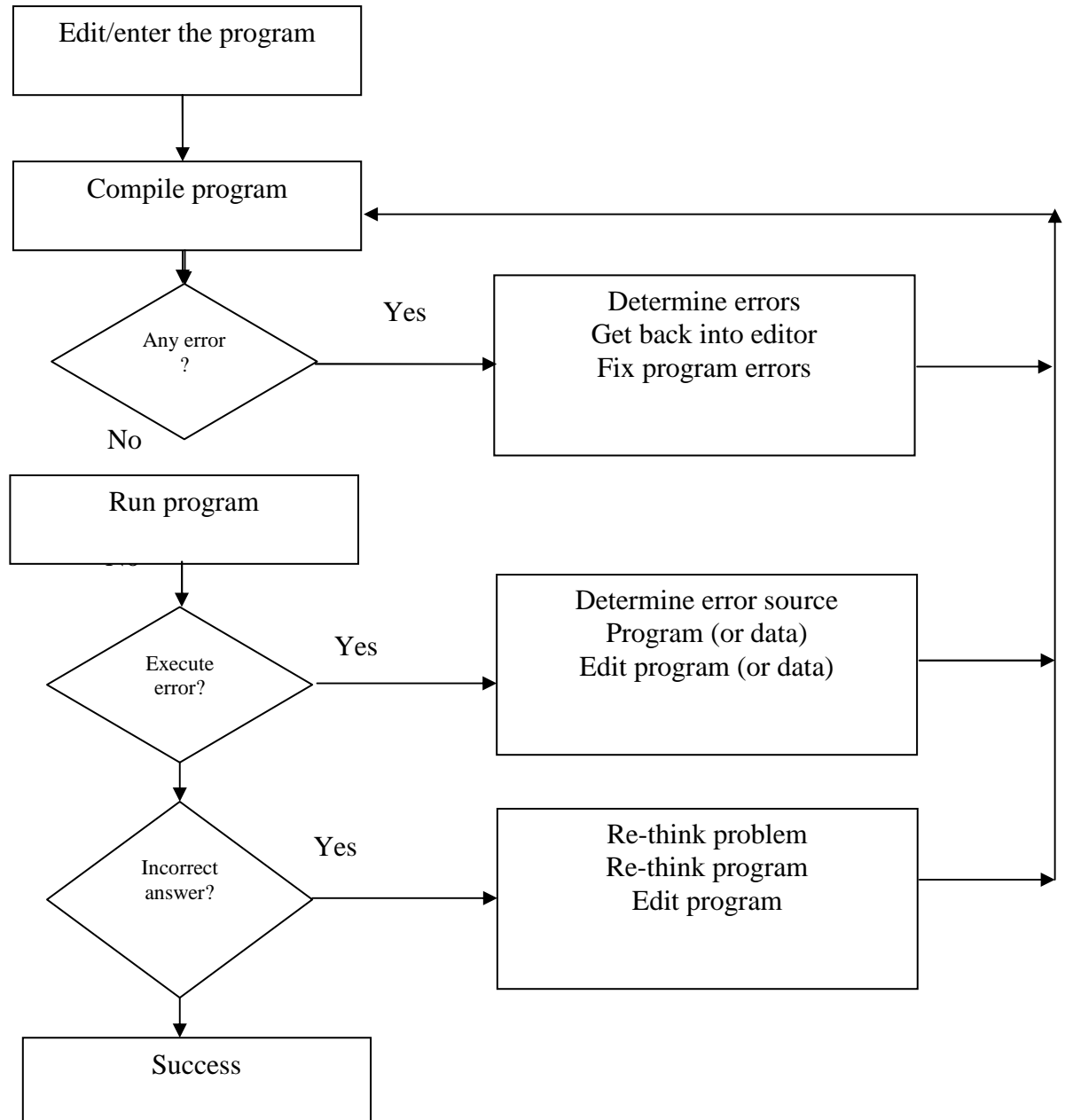
```
For I = 1 To 10
  For J = 1 To 10
    For K = 1 To 10
      ...
    Next J
  Next I
Next K
```

---

- Line 1: There is no bug
- Line 2: Is a nested loop, no bug
- Line 3: Also a nested loop, no bug
- Line 4: There is a bug - **syntax error**. This suppose to close the inner loop on line 3hence, **NEXTJ** is not appropriate. It should be **NEXT K** to close the inner loop.
- Line 5: Similarly there is a bug - **syntax error**. This suppose to close the inner loop on line 2 hence, **NEXTI** is not appropriate. It should be **NEXT J** to close the inner loop.
- Line 6: There is a bug - **syntax error**. This suppose to close the outer loop on line 1hence, **NEXTK** is not appropriate. It should be **NEXT I** to close the inner loop.

Similarly Note, if you omit counter in a **Next** statement, execution continues as if counter is included. If a **Next** statement is encountered before its corresponding **For** statement, an error occurs.

2. The Students at the end of week 12, should be able to illustrates test and fix bugs approach of debugging a program.



## WEEK 13

### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ How to write simple structured program

#### To write simple structured program

During this practical we want to display the numbers from 5 to 20 and numbers from 15 to 5. The main form look like this form

The image shows a Windows application window titled "Looping". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is divided into two sections. The top section is titled "Do Loop" and contains a large yellow rectangular text area. Below the text area are two buttons: "Start for loop" and "Clear text". The bottom section is titled "For Next Loop" and also contains a large yellow rectangular text area. Below this text area are two buttons: "Start Do loop" and "Clear text".



1. Open project week 13
2. Double-click the " Start Do loop" Command button and write this code

```
Dim counter1 As Integer
```

```
counter1 = 5
```

```
Do While counter1 < 21
```

```
    Label1.Caption = Label1.Caption + " " + Str(counter1)
```

```
    counter1 = counter1 + 1
```

```
Loop
```

3. Double-click the " Clear Do text" Command button and write this code

```
Label1.Caption = ""
```

4. Double-click the " StartFor loop" Command button and write this code

```
Dim counter1 As Integer
```

```
counter1 = 5
```

```
Do While counter1 < 21
```

```
    Label1.Caption = Label1.Caption + " " + Str(counter1)
```

```
    counter1 = counter1 + 1
```

```
Loop
```

5. Double-click the " Clear For text " Command button and write this code

```
Label2.Caption = ""
```

6. Save your project with the name "week 13" in your folder.
7. Run the program

## WEEK 14

### SPECIFIC LEARNING OUTCOMES

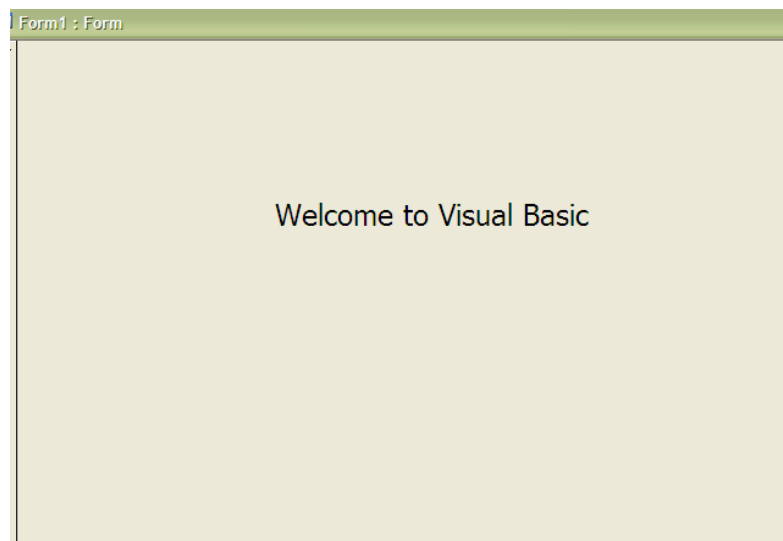
#### To understand:

- ✓ How to write simple structured program

#### To write simple structured program

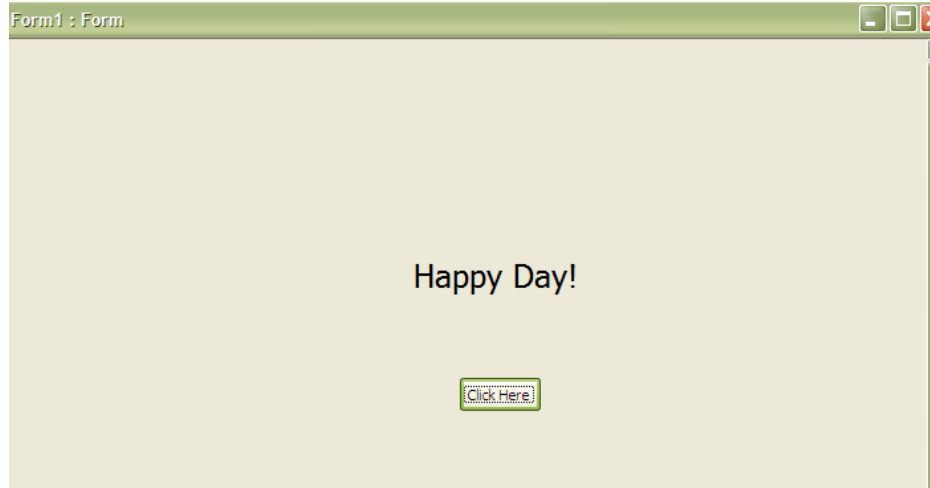
During this week students are to create a new project in VB with the name “week15” and the final form look like the following figure with “Welcome to Visual Basic”.

1. Create new project
2. In the New Project box, Click Standard.EXE and then Click OK.

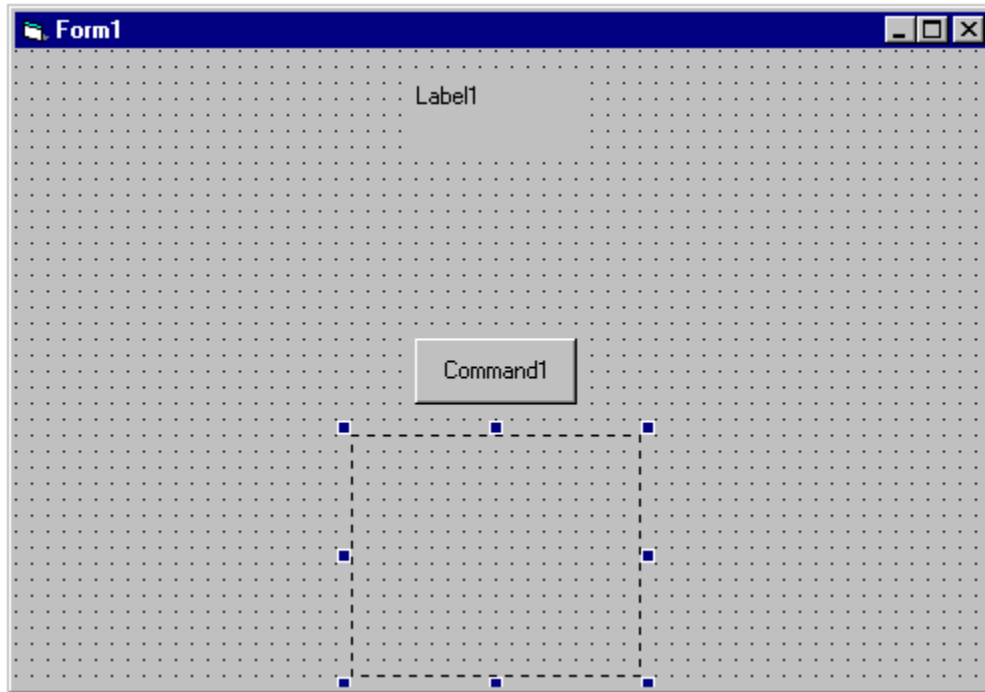


3. Add Label controls and change its properties to look like this form.
4. Double click the Label with the caption “1”, write this Code in the click event procedure  
Label1.Caption = “**Welcome to Visual Basic**”
5. Save your project with the name “week15” in your folder.

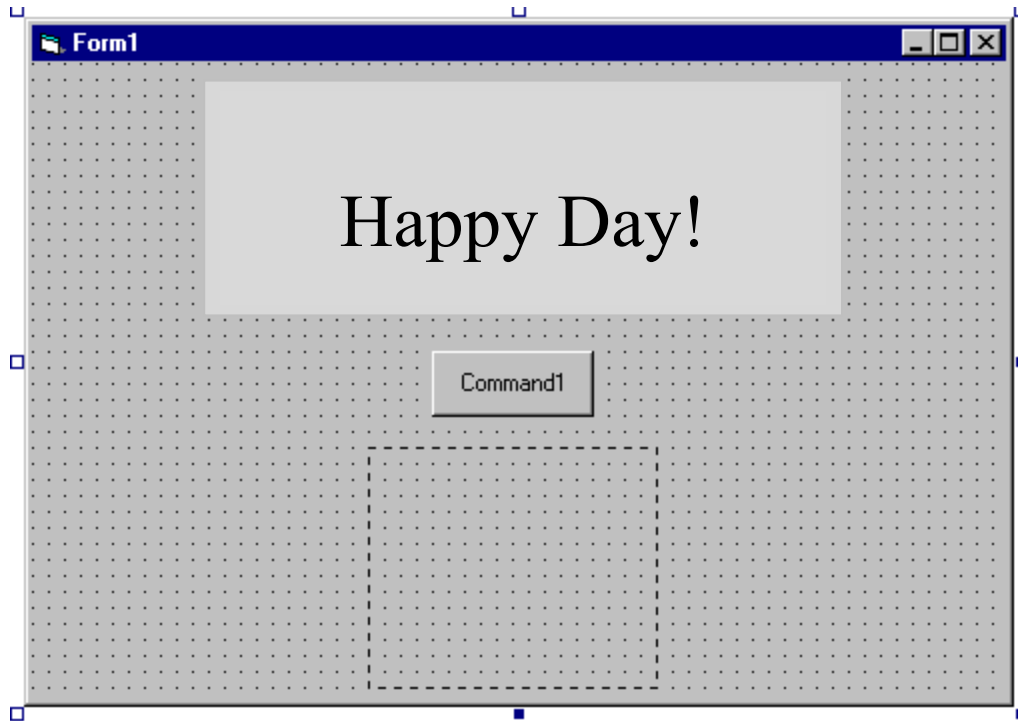
- b. During This Lab we want to create a new project with the name “week14” and the final form look like the following figure and when you click on the button "Click Here" the text on the form will change with “Well Done”.



8. Start Visual Basic 6
9. In the New Project box, Click Standard.EXE and then Click OK.
10. Double-click the Label control. Visual Basic puts a new label in the center of your Form window. Move the control higher on the form to the approximate
11. Double-click the Command Button control to place a command button in the middle of your Form window. Leave the command button where it appears for now.
12. Click once on the Image control. Move your mouse to the Form window. The mouse pointer appears as a plus sign to show that you've selected a control.
13. Drag your mouse to outline the new Image control on the Form window. Next Figure shows where you should place the control. Try to get close to the figure's placement and size.

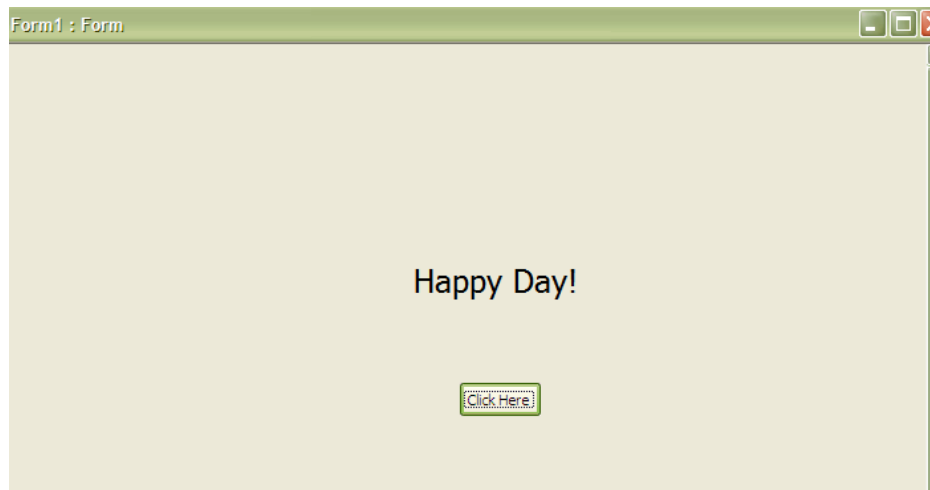


14. Click the control named Label1 to display the label's properties in the Properties window.
15. Click the Properties window's Caption entry.
16. Type Happy Day! As you type, you'll see the label's caption change.
17. Click the Properties window's Font entry. A font name doesn't appear, but a set of ellipsis does. Ellipsis after a property value always indicates that you can select from a list of choices rather than type the value directly into the Properties window.
18. Click the Font entry's ellipsis to display the Font dialog box. A font supports multiple properties, so you must use the dialog box to specify the font name, size, and style.
19. Select the Courier New font, with a Bold style and a 36-point size.
20. Close the Font dialog box. Resize the label's caption to see more of it. Although you can drag the label's edges, you'll instead enter location and size values directly in the Properties window so that your application matches the one in this book.
21. Set these Properties window values: Left, 1,320; Height, 1,695; Top, 120; and Width, 4,695. You can now read the new label's caption, but the caption is still left-justified. Expand the Properties window's width if you need more room to read the property values.
22. Click the label's Alignment property. Click the drop-down list box arrow to open the list box and you'll find three values: Left Align, Right Justify, and Center Justify. Select Center Justify, and the label straightens up nicely.



23. Select the form's command button, click the Caption property, and then type Click Here for the caption.
24. Double click the command button, write this Code in the click event procedure  

```
Label1.Caption = "Well Done"
```
25. Save your project with the name "week15" in your folder.



## WEEK 15

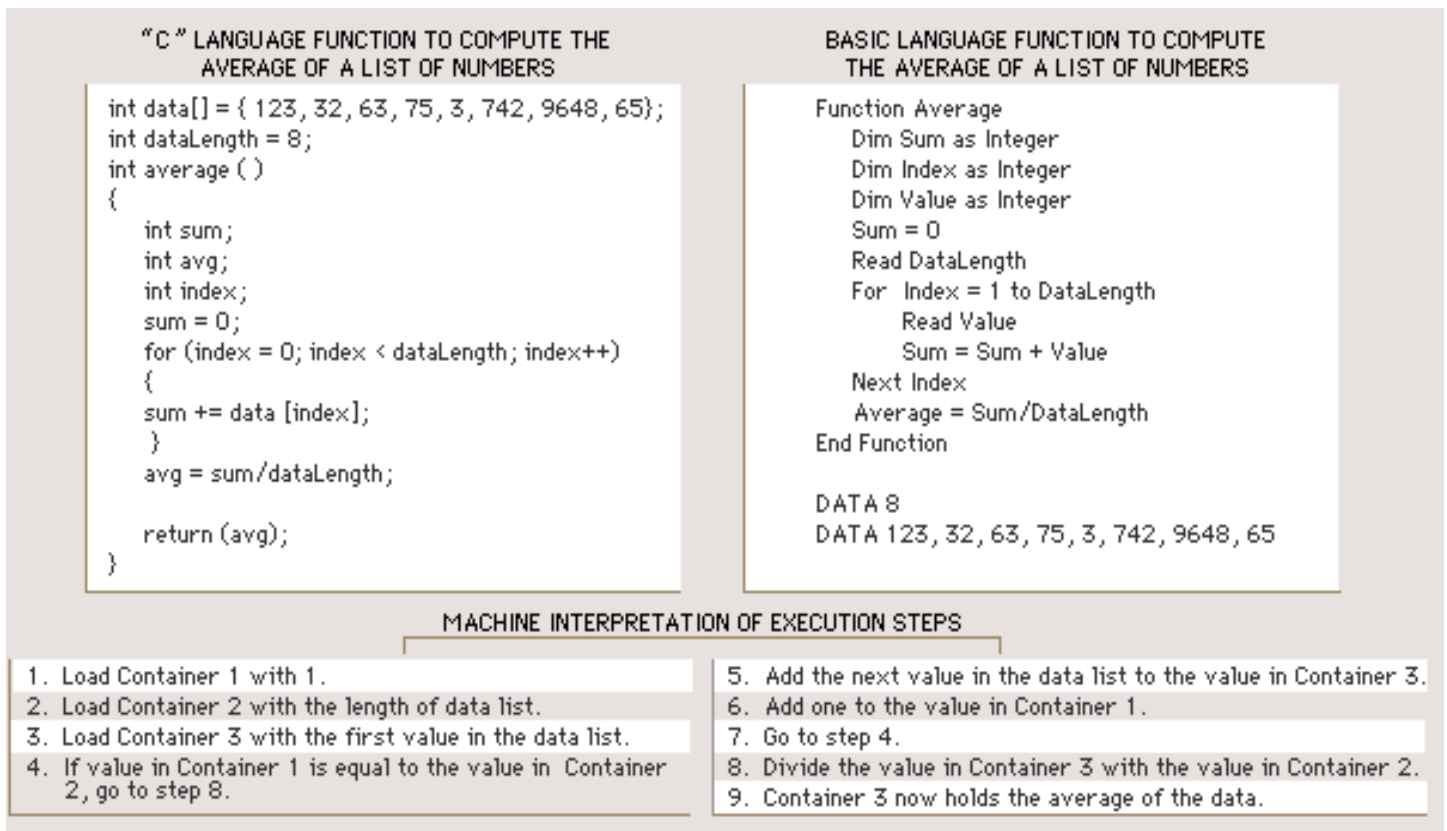
### SPECIFIC LEARNING OUTCOMES

#### To understand:

- ✓ To be able to identify properties, events, objects and class in a running OOP

To identify properties, events, objects and class in OOP

At the end of this week 15, student should be able to write and run simple program in C++ language and BASIC language to compute average of a list 8 values as shown in the diagram below:



#### Students Assessment

1. Students are study the program editors of VB and C++ and state the difference and similarities between them.
  - i. \_\_\_\_\_

- ii. \_\_\_\_\_
- iii. \_\_\_\_\_
- iv. \_\_\_\_\_
- v. \_\_\_\_\_

2. Which of the object oriented programming languages do you find more robust and comfortable? Give 3 reasons for your answer.

- i. \_\_\_\_\_
- ii. \_\_\_\_\_
- iii. \_\_\_\_\_