BJP -

RSS

Page No.
Date

VB6 ① VB Net

Steven Holzner
James Powell
Sam - Black Book

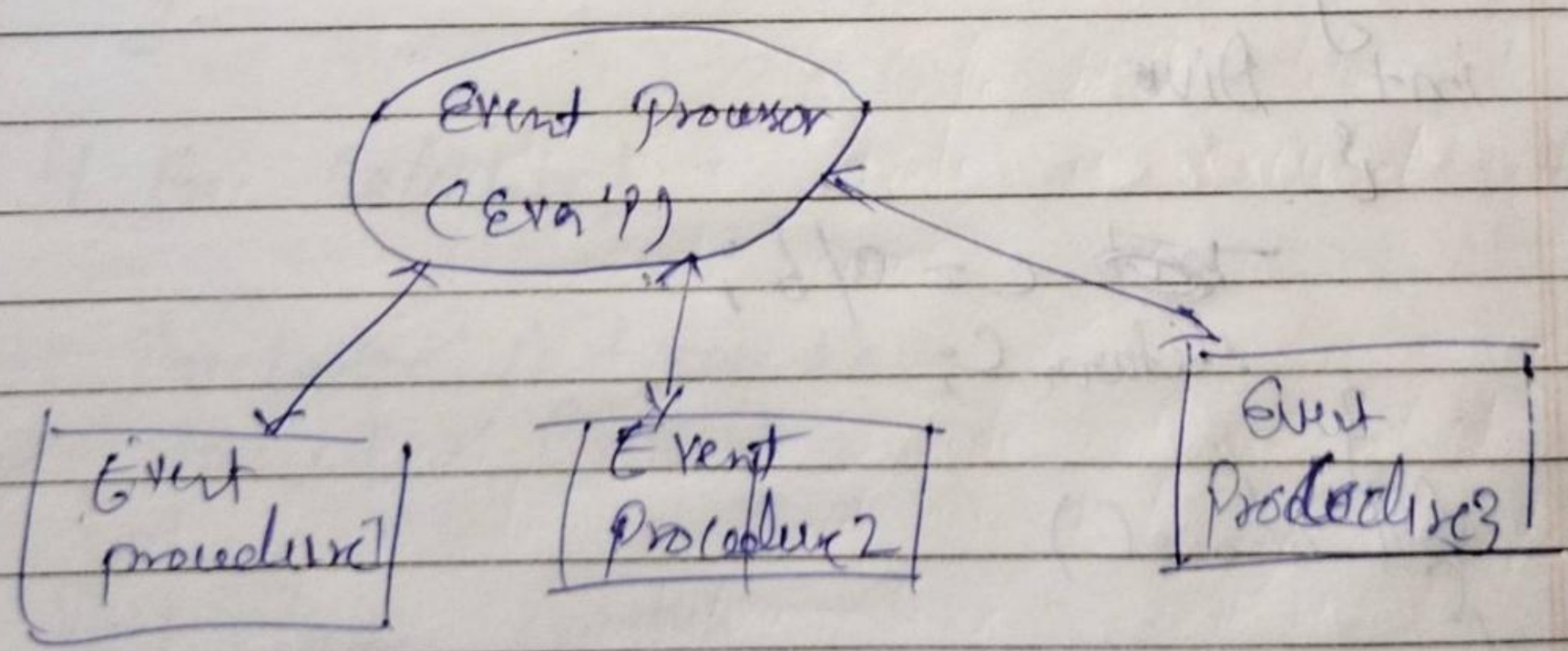
Visual Basic / (1960's Dartmouth College)

B = Beginners
A = All - purpose
S = Symbolic
I = Instruction Code
C = Code.

Visual Basic - 1991 - 16 bit

Allow To develop Windows, GUI based Application
It is event driven language that is Code Remain Idle until called to response to some event
Visual basic garanteed an event processor. Once (Name event) a event is detected record corresponding that event called word processor is excuted, And program control is return back to event processor

Event driven




Scanned by TapScanner

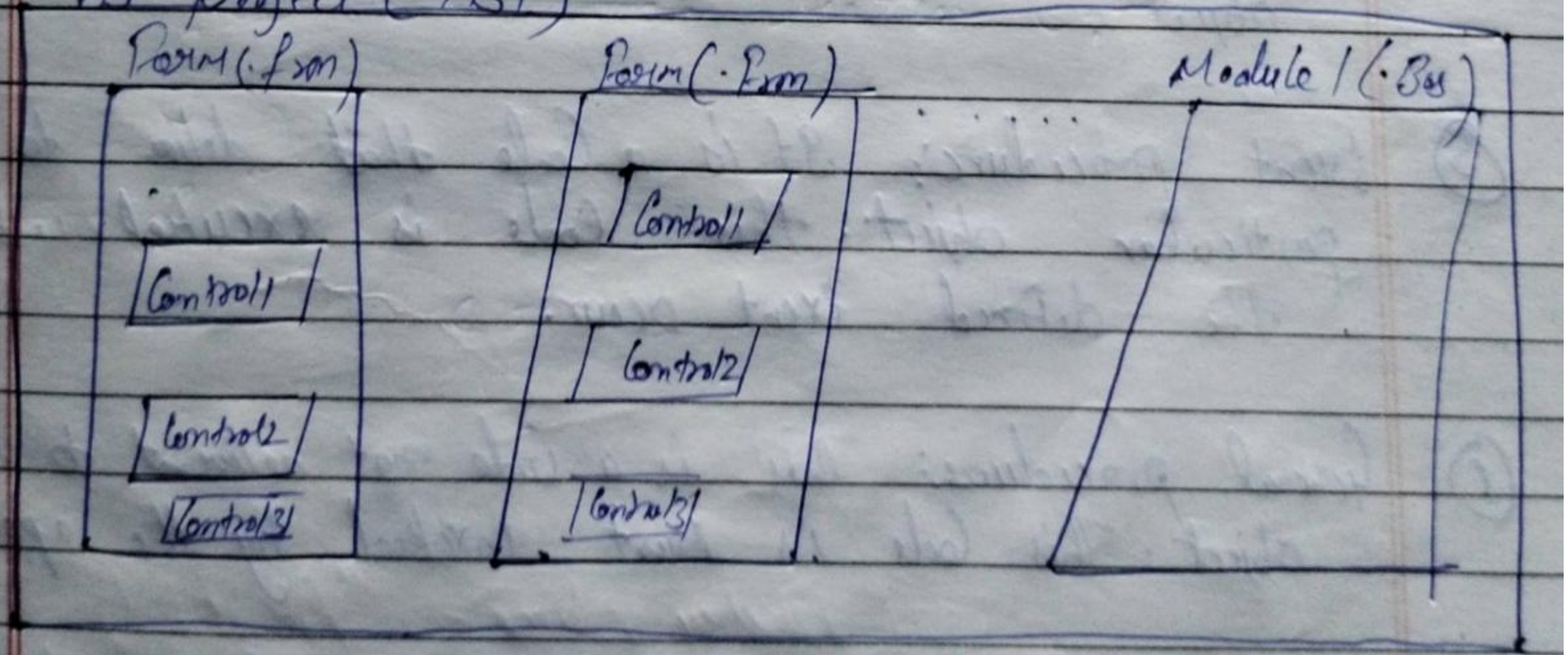
Page No.
Date

## Features of Visual Basic

1. It have full set of object to draw VB application
2. It Contain lot of Icon and picture to Use
3. It response to key board and Mouse action.
4. Clip board and printer acess.
5. It support Mathematical, strick handly and Graphic function.
6. It Can handle fixed and dynamic Variable and Control array.
7. sequensal and random acess file support.
8. It have powerfull database access tool.
9. It have usefull debugged error handleing
10. It have activace support
11. Actual and development Windows numbers distribution of your application simple.

17/8/2019

## Structure of a VB application.




Scanned by TapScanner

A VB project defined by VBP and application and project is made up of

① Form ⇒ A Form is a window for User interface

② Control ⇒ Control are the graphical feature drawn on form to allow User interaction.

i Text box

ii Labels

(iii) Scroll bar

(iv) Command button

(v) Form and Control are object

③ Property : Any characteristics of a form or a Control is specified by a property

① Name ② Caption ③ size ④ Colour ⑤ position

~~Property~~ Initially VB applied default property which we can change at design time or run time

④ Method : These are built in procedure that can be invoked to impart sub action to a particular object

⑤ Event procedure : It is a Code that delion to a particular object. this Code is executed when the defined event occur.

⑥ General procedures : This is a Code not relevant to any object. This Code is must invoked by the application.

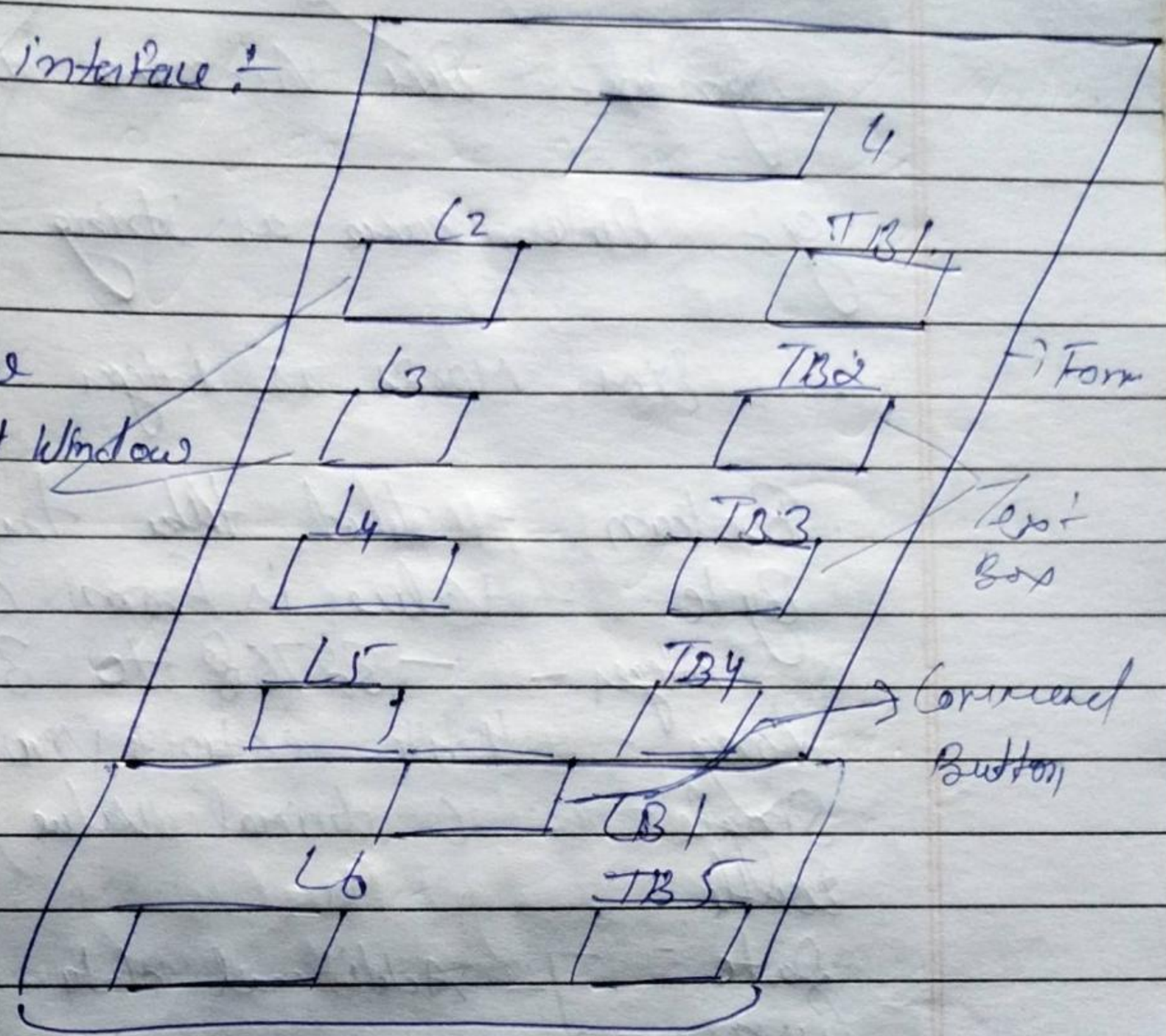
⑦ Module : It is a Collection general procedure Variable declaration and Constant declaration by the application.

Scanned by TapScanner

Step of developing an application

Designing User interface :-

(i) Form Window
(ii) Tool box
(iii) Property Window
(iv) ~~Tool box~~ Product Window
(v) Label




(4) Setting properties :-

Page No.
Date

## Variables :-

Keyword

Syntax :- Dim Var-name as type

eg :- Dim name as string

Dim Marks as Integer

Boolean - that take True and False
Byte - Value is from 0 to 255
integer - -32768 to 32767
long - 4 bit extend version integer
Single - 6 decimal value input
Double -
Date - ] - Additional data type
Currency -
Variant :- Data handling

User defined

Start of User defined type

Type St detail
Dim Name as string
Dim RollNo as Integer
Dim Marks as Single

End type
End of type defination

Dim Akshay as st detail

Akshay Name = "AKSHAY"
Akshay RollNo = 5
Akshay.Marks

Array - Variable

Scanned by TapScanner

Page No.
Date

10 byts
4 byts
4 byte

Std detail
18 byts.

Scanned by TapScanner

Page No.
Date

Array implementation:

C++ Programming

Maxsize = no of elements.

SP = -1

Push  Pop

(1) OverFlow => Full of Stack (Insert)
(2) UnderFlow.

```
If (Stack == Maxsize - 1)
{
printf : OverFlow
}
else {
        SP = SP+1;
        item[SP] = UnderFlow.
}
```

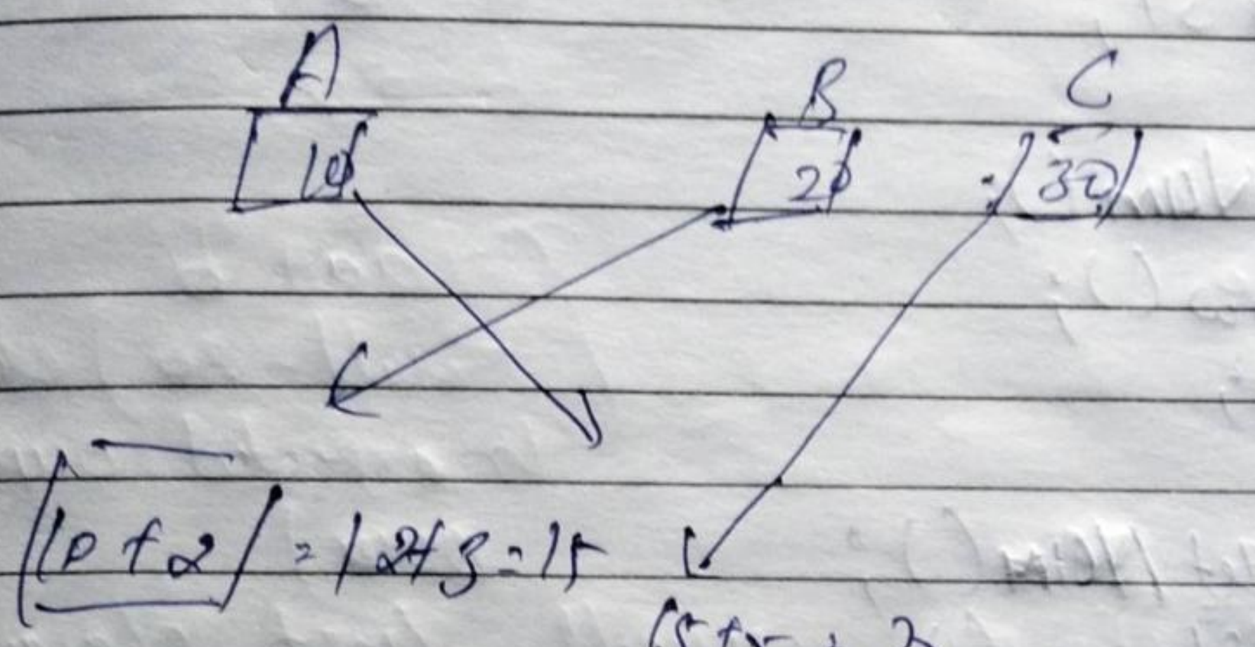
Pop

```
if (SP == -1
{
printf - UnderFlow
}
else
{
Value: item [SP];
delete item
}        SP == -1;
```

Scanned by TapScanner

Page No.
Date

# Static Data Member




Static int x; // Default = 0

Class Sample
{
Static int x;
};

int Sample :: n = 10; // Defination

void alum()
{
Cout << " Enter a number
Cin >> a;
x = x+1;

Show()
Cout << " number = " << num;
Cout << " Static x = " << x;
}
};

Scanned by TapScanner

Void Main ()
{
Simple S1;

S1. get Num ();
S1. Show ();
Sample S2;

S2: get Num ();
S2. Show ();

Sample S3

S3. (get Name ();
S3. Show();

3.

Output
Enter a Num=10
Num=10
Static X = 1

ENTER-A NO=20
Num=10
Static X=

Scanned by TapScanner

Page No.
Date

Type Declaration Character

% Integer
& long
! Single
# Double
@ Currency
$ String

e.g 1. Dim name as string
~~Dim~~ or
Dim name $

2. Dim RollNo as Integer
Dim RollNo %.
3. Dim Marks as Single
Dim Marks!

Type of Declaration

1.) Implicit
2.) Explict
3) option Explict → tool.
↓
option
☑ Require variable declaring.

1) Implicit: In implicit declaration, you can use variable directly in program without declaring it. When you use a variable name the VB automatically creates a memory area of type variant.

Scanned by TapScanner

| Page No. | |
|---|---|
| Date | |

e.g I

a = 10
b = 20
c = a+b
Result = 30

II

a = 10
b = 20

c = aa+b
(Result = 20) etc.
(Unexpected output without error)

In eg II if we misspell first character a or double a (aa) then it will not give error rather it declares a 4th variable with name aa, and the result will be unexpected without any error.

2) Explicit :- In explicit declaration it is necessary to tell VB that a variable can not be used if it is not declared.

3.) Option Explicit -> tool
↓
option
↓
[V] Require Variable declaration

Constants (do it yourself)

Scanned by TapScanner

Page No.
Date

# Scope of Variable

1. Global
2. Local
3. Static

Integer a As Double

[ ≡ print a; a++;

[ Integer b As Double
print a, b; a++; b++; ]

a = 10
12
13

[ Static Integer c As Double
print a, b, c; a++, b++, c++; ]

b = 10
= 11
b = not accessible

[ ≡ a = 13
b = 0
c = 13

c = 12
c = 13

Scanned by TapScanner

Page No.
Date

Print Statement

a=10 print a
b=20 print b.
c= a+b.
print c

print a, b, c — COMA form
print a; b; c. On line → Semicolon

print tab(10) a; b; c;
[There space tab(10) is between output of a b, c.

Arithmetical Integer

+
−
*
/ → Decimal \ → Integer.
^
Mod
<> Not equal to

print c
C= a*b
print
C=a/b
print C
a= a^b

Boolean operator

Scanned by TapScanner

Page No.
Date

① AND
② OR
③ Not
④ XOR
⑤ Eqv

l, T ⇒ To use the Compiler too string values.

VB Branching

Syntax IF
Then statement
End IF

1) IF/ Then

eg:-

IF a>b
Then
print " a is greater."

eg:-

IF Balance - cheque amount < 0
Then
print " You are overdrawn, cheque cannot be cleared".

⇒ ② If a>b And a>c
Then
print " a is Greater than b & c both".

Scanned by TapScanner

Page No.
Date

1) If and only If : This Work for Case When If and only If give Cond.
2) Single line statement : This Work only single line Statement
3) This will Work only when Cond- is true otherwise it will not print or do anything

If then End if

Eg:

```
If a>5
Then
    print" a is greater"
    print" We will use a for calculate"
    print" Thankyou"
    End if
```

If / Then / Else / End if

Syntax:

```
If condition
Then {
Else {
End if
```

(Greatest of 2 Nos)

If / Then Else if / Else / End if.

Scanned by TapScanner

Page No.
Date

Assignment

1. VB requirement
2. Welcome message print
3. To find sum of numbers.
4. Calculate Marks of student display grade
5. Calculate the salary of Employee
6. (

I. If a>b
Then
print " a is Greater"
Else
print " b is Greater"
End if

IV. If a>b
Then print " a is b
Else if
a<b
print " b is Greater"
else
print " a & b is Equal"
End if.

Scanned by TapScanner

20.Sep 2019.

## IF / Then / Else if / Else / End if

```
IF age < 5 then
    print " child"
Else if age >= 5 And age < 12
    print " Toddler"
Else if age >= 12 And age < 18
    print " Teenager".
Else if age >= 18 And Age < 45
    print "Young".
Else if age >= 45 And age < 60
    print "Middle age".
Else If age >= 60 Then
    print " Senior citizen"
Else
    print " Every one Else."
```

Output

Age

Enter age [ ]

Text Box [ ] Find Categ.

## Select Case

Syntax:

```
Select Case    Case Variable
  Case: —
  Case: —
  Case —
  Case Else
    —
End select
```

Scanned by TapScanner

Select Case Age

Eg:- Select Case Age
Case 0 to 5
print "Child"
Case 6 to 11
print "Toddler"
Case 12 to 17
print "Teenager"
Case 18 to 44
Print "young"
Case 45 to 59
print "Middle aged"
Case Else.
print "Senior Citizen".

☆ loops - D

① For - Next → It executes set of statement certain no. of ~~element~~ times.

Syntax :-
For Counter = Start to end
Next Counter

e.g:- For n = 1 → initialize to 10
~~print~~ = n
Next n

Output
1
3
5
7
9

Step is used to move forward or backward by certain number

(2) For Each Next →

Syntax:

For each element in group

Next element

eg:- For Each a in a[i][j]
print a[i][j]
Next a

eg:- For each a, b, in a[i][j], b[i][j]
c[i][j] = a[i][j] + b[i][j]
print c = c[i][j]
Next c

(2) While - Wend:

Syntax

While bool.

Wend

eg:- DIM n as Integer
n = 1

While n <= 20
print n
n = n+1
Wend

(1) W.A.P to find factorial of a no
(2) W.A.P to find square of a no.

Scanned by TapScanner

Page No.
Date

Loops

Do - loops

I → Do - While loop

II → Do loop while

III → Do Until loop

IV Do loop Until

Do While Cond
loop

Do
loop while/ Cond Until

n
n%2 | - 50 print
True
else odd

File to be checked from 21-23 Oct 2019

Arrays:

Index: 0

| a | a+1 | a+2 | a+3 | | a+n |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | |

start

Declaration of Array:

Syntax:

Dim arr-name(Size) as Type

eg: Dim Arr.(a) as Integer

No. of Elements = 10

0 - 9

Scanned by TapScanner

Page No.
Date

Option Base 1
Dim Arr (9) Integer
Now element are
9

→ 1–9

II Dim (1 to 10) as Integer
Range (1-10)

| | | |
|---|---|---|
| 1111 | 1111 | 1111 |

| A1 | A2 | A3 |
|---|---|---|
| 1 — 10 | 11 — 20 | 21 — 30 |

Array
Dim A (9) as Integer
For
a = 0 to 9
print A(a)
Next a.

MultiDim - Array

Syntax
Dim Array name (rows, Column) as Type
eg:
Dim (Arr (9,9) as Integer
10 X 10
option Base 10
Arr (15,15)
6 rows

Page No.
Date

Arr (1to10, 1to10)
10 X 10
Arr (9, 9, 9)
Arr (9,9,9,9)
Arr (1to10, 1to10, 1to5)

Size of Array

Arr (9, 9, 9) as Integer
10X10X10X2 = 2000

Array DT = 20
+ 4X3 =
Dim

Dim
20+4+ (11X8) = 112
20+4X2 + (6X6X8) =

Function
Is Array
Counted
Unbounded

Scanned by TapScanner

Page No.
Date

# I/O with Input Box() and Msg Box()

① Msg Box → Function display the Msg to the user and gets a return value to one of the button in Message box: it is like a dialog box used for output during program execution.

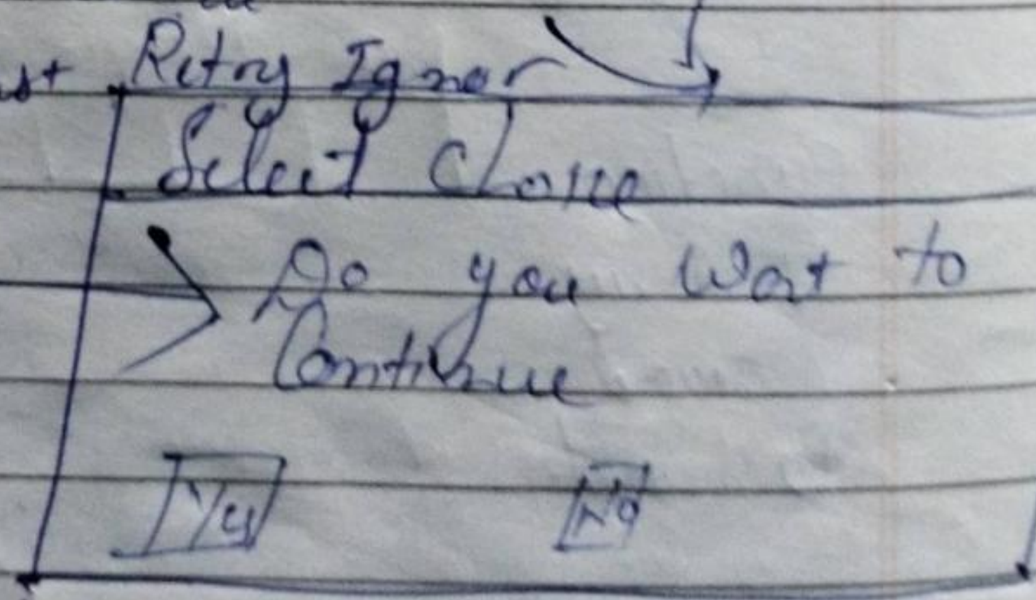
Syntax:-

Msg box [ StringPrompt, [Button], [Title] ]

Here string Code argument : String display as Message in dialog box having maximum length of 1024 character.

The button is optional argument the specific button to put in Message box
By default button value is zero (0).

| Value | Constant | Description |
|---|---|---|
| 0 | Vb ok only | |
| 1 | Vb ok Cancel | |
| 2. | Vb Abort Retry Ignor | |
| 3. | Vb yes No Cancel | |
| 4. | Vb Yes No | |
| 5. | Vb Retry Cancel | |




Scanned by TapScanner

5 4 → 2a

Page No.
Date

Default Button.

Return Value.
1. OK
2. Cancel
3. Abort
4. Retry.
5. Ignor
6. Yes
7. No.

Tittle → The title bold string to the display In title bar in the dialog box.

Syntax:-
Choice:- Msgbox ("Do you want to Continue", Vb Yes, No or "4", Select choice")

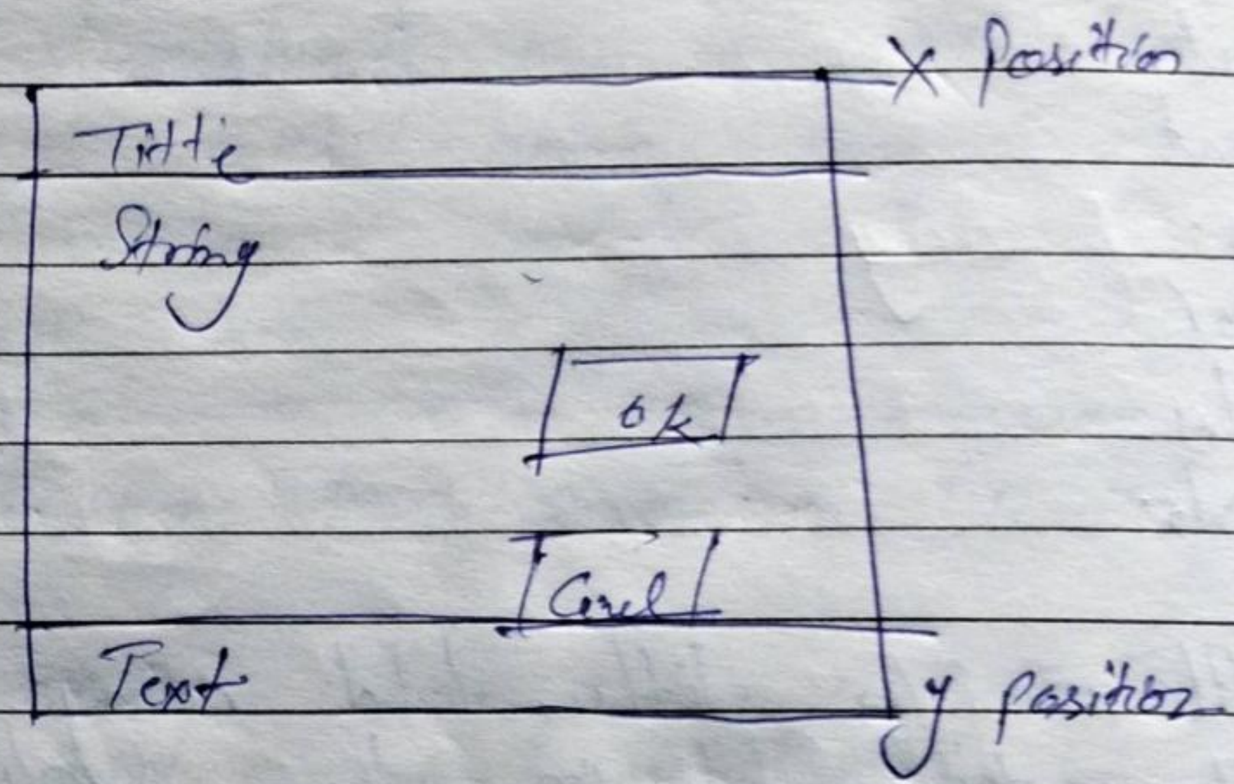
Icons
Vb critical X
Vb Question ?
Vb Exclamation !
Vb Information

Input Box.
① It is used to get string from the User. It display ok and Cancel Command Button

Page No.
Date

Syntax:

Input Box (Start prompt, title, default, x pos, y pos)




Working with Function

① Hiding and showing Form

Procedures:

① Sub-procedures: performs a valdetine task but
② it do not return a value.
② General purpose: Subprocedure (Sub routine)
i Event procedure
↳ A General procedure performs a well define task has it own name and it define separatly. It is
ii Called by an Event procedure and general procedure.



Scanned by TapScanner

Page No.
Date

Syntax:

I public / private Sub procedure-name [Any-1]

End Sub

Tools

procedure

Add procedure

Calling sub routine / General purpose

~~Calling~~

Using Call statement

without " "

→ Call pro name

pro-name Argument list

eg: public Sub Addition.

Dim a as Integer

" b " "

" c " "

a = Inputbox ("Enter First No")

b = Inputbox ("Enter 2nd No")

c = a+b

Msgbox ("Addition = " & c)

public Sub formload()

Scanned by TapScanner

Addition
Call Addition
End sub

Call of
Subroutine

(11) Event procedure: VB application is responses to an event such clicking a Command button and these event are associated with both form and Control a Code that run a response to a click or event is called event procedure or routines. An event procedure are always private

Syntax:

```
T1, T2, T
private Sub Control-name -
        Event name ( )

End Sub
```

eg:

```
Private Sub Comm btn - click ()

T1 = a
T2 = b
T3 = c

End Sub
```

T1, T2, T3
Text box

Scanned by TapScanner