

Imp

## \* Exaption / Error Handling :-

A program error comes in three variety.

- 1) Compile Error.
- 2) Run-time Error.
- 3) Logic Error.

### 1) Compile Error :-

V. Basic attempt to convert our project code to Machine code language Code ~~just~~ <sup>during</sup> compilation. We get an error when we break Syntax (rule) of V.B.

For example:-

Try to spelling ~~End~~ as Quit or Stop. The compiler can ~~only~~ translate the exact spelling of words, we also receive a compile error if we use wrong punctuation in wrong place.

For example:- ~~Wrong punctuation~~

Command 1, Caption = "Add"

instead of Command 1. Caption = "Add".



## 2) Run-time Error :-)

If our project / program Halts (रोकती) <sup>Stop</sup> during execution that is known as Run-time Error. In that case V-Basic displays a dialog box goes into break time & highlights the statement causing the problem. That statement cannot execute correctly so it causes run-time error. Run-time error can be caused by attempting to do impossible arithmetic operations. Such as

- Divided by Zero.
- Square root of -ve number.

## 3) Logic Error :-)

With Logic Error our projects run & produces incorrect result  
For Ex:-

If we want to add two numbers & we get the difference of two numbers as a result. This type of Error is known as Logic Error.

## • Exception Handling :-)

An exception is



generated at the run-time error. ~~when~~  
It Halts the program execution.

An exception can be handle with the help of Exception Handling Mechanism.

~~When~~ When a Run-time Error occurs V-Basic generates an error number & check it against a table on ~~an~~ error code. We get ~~a~~<sup>the</sup> error number & set proper action without terminating the process (program) the statement used in error handling are:-

- On Error Statement.
- Error - Object.

1) On Error Statement :- (Error Handling Mechanism).

On Error Statement trap (catch) the error for on-error statement we must do the following thing:-

- i) ~~Terms~~ on the Error-Handling Feature use in On-Error Statement.
- ii) Create Error-Handling Code routine which are set of our other code with line-label.



iii) Determine how & where the program is continue after an error is taken care of.

28/8/12

### 1) On-Error Statement is

The On-Error Statement must be placed at the beginning of any procedure where error may occurs.  
(It is on the Error Handler).

There are 3 forms of On-Error Statement

- 1) On Error Goto Line-Label.
- 2) On Error Resume Next.
- 3) On Error Goto 0.

### 1) On Error Goto Line-Label is

On Error Goto specify the label of line where our code for handling the error are present. The Line-Label itself followed by column.

Note:- The error handling code must be in



error.

Syntax:-

On Error Goto 0

Example:-

```
Private sub Command1_Click()
    On Error Resume Next
    Print "Jyoti"
    Print 8/0
    Print "Deepika"
    On Error Goto 0
    Print "Kheshi"
    Print 7/0
End sub
```

\* Err-object (Pre-defined Object)

The Err-Object holds the information about the error that has occurred. We can check the property of error object to determine the error number & the description of the error or the name of the object or application error that causes error.



is stored in source property. It contains the error number ranging 0-65535. We don't define or include err-object. It has Global Scope & it will automatically we call in any part of our project. Some error number & it's description are given below:-

<u>Err. Number</u>	<u>Err. Description</u>
7	Out of Memory.
9	Subscript out of range.
11	Division By Zero.
13	Type Mismatch.
58	File Already Exist.
75	Path /file access Error.
482	Printer Error.
24574	Font Not Installed

Private Sub Command1\_Click()

ON Error Goto 1

Print "Jyoti"

Print 2/0

Print "Deepika"

1: MsgBox ("Error is handled" & "ErrorNo=" & Err.Number & "Error will be=" & Err.Description & "Error.Source=" & Err.Source).

Teacher Signature