Error Types and Debugging

Error Types

There are three types of error that can occur when writing and testing a program.

Error Type	Description	Example
Syntax Error	The rules of the programming language have been broken. The program **will not** start.	FOR index **IS** 1 to 10 • is should be = IF age = 5 **THEEN** • Theen should be then
Execution Error	Program is asked to do something impossible or illegal. The program **will run** but will **crash**.	answer = total / 0 Set age TO “Fred”
Logic Error	Program will run and will not crash. Does not produce the expected results.	SET answer TO 6 * 4 SEND “6 + 4” = answer • Expected result is 10 but error in the first line means 24 is displayed.

Debugging

Debugging is the process of finding and correcting errors.

Some types of error are easier to identify than others because the programming environment will help.

- The program will **not run** at all if there is a *syntax* error
- The program will **stop running** if an *execution* error is encountered

Debugging

Logic errors are more difficult to find because the program **will run** but will produce incorrect results.

There are a number of debugging techniques that can be used to identify logic errors.

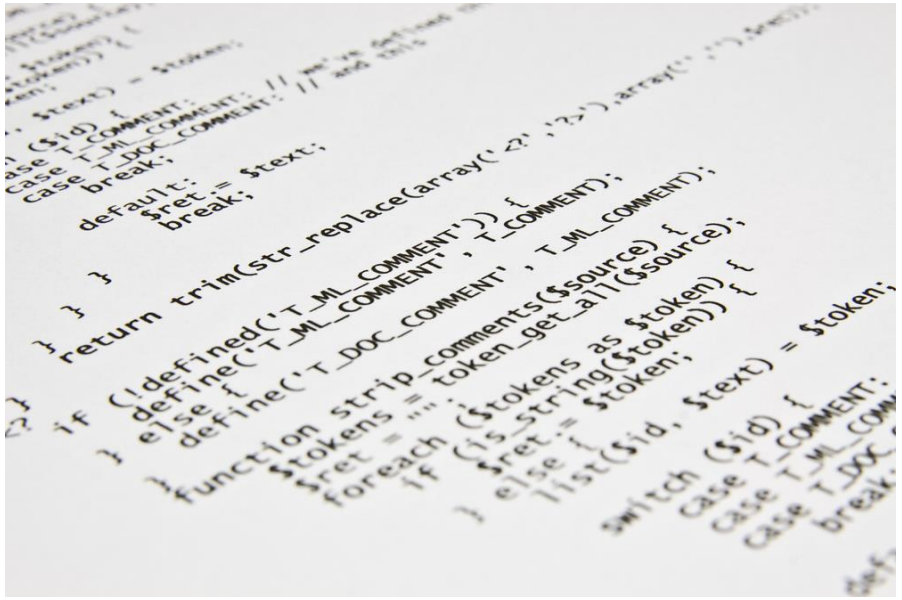
- Dry Run
- Trace Table
- Trace Tools
- Breakpoints

Dry Run

A Dry Run involves **manually** stepping through each line of code using test data.

As lines of code that make changes to variables are reached, these changes are recorded using a **table**.

This should highlight positions in the code where variables are changing to unexpected values.



Trace Table

A trace table is similar to the table used to record variable values during a dry run.

Trace is often used to record the changes to variables when testing an algorithm for a specific sub-program.

A trace table allows the tester to check the result of a number of different values of a variable.

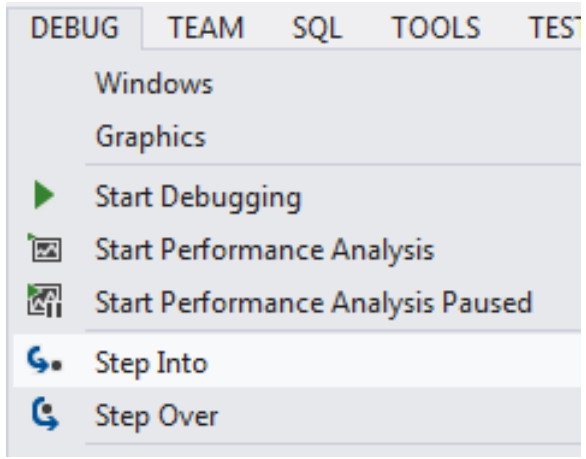
	lower	upper	middle
1st pass			
2nd pass			

Trace Tools

Trace tools are a debugging feature of some programming environments.

Trace tools allow the program to be executed but the programmer can **step through** one line at a time.

```
If lblQuestion.Text = 1 Then
    If RadioButton2.Checked = True Then
        lblScore.Text = lblScore.Text + 1
        MsgBox("Correct")
    End If
```



Trace Tools

This lets the programmer view the line of code being executed.

A **watch** can also be set to allow the programmer to view (and record) the values of a variable as it changes

```
Dim score As Integer

score = 5
score = 10
score = 15
```

Watch 1

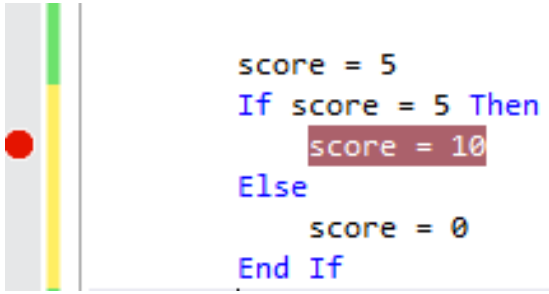
Name	Value
score = 10	True

Breakpoints

Breakpoints are another debugging feature of some programming environments.

Setting a breakpoint sets a point in the code where the program will stop execution.

Breakpoints can be set to stop executing at a particular line of code, or when a variable reaches a particular value.



Once the program has stopped, the values of variables can be examined.