

KNOWLEDGE ORGANISER :: DEBUGGING AND TESTING

DEBUGGING AND TESTING	
Trace Table	An offline method of tracking the values of variables through the running of a procedure
Overflow Error	An error produced when a number becomes longer than the number of bits allocated to it. The extra bits are lost.
Logic Error	An error with code where it compiles correctly but produces incorrect results
Syntax Error	An error with the code where the computer can not recognise it as code
Runtime Error	An error which occurs during operation of the program, not during compilation

Syntax Errors can be diagnosed by **Compilers** and **Interpreters**
 If syntax errors exist in the code the compiler and interpreter will be unable to turn the code into source code
Logical Errors are more difficult to spot and correct
 Compilers and Interpreters cannot detect logic errors
 Logic errors can only be spotted using a Test Plan

TYPES OF TEST
Functional Testing is used to test the user interface and spot logical errors
 The purpose of these test is to see if the program meets the original requirements
 It should not be left until the end – functional testing should be completed as the solution is being developed
Performance Testing is used to test how quickly the programs runs
Usability Testing is used to test the user friendliness of the solution
Security Testing is used to test the security of the solution
Load Stress Testing is used to test how the solution copes under extreme conditions

EXAMPLE
 Jerry has written the following function
 It multiplies a given positive integer by all the positive integers less than it
 For example – if the integer was 5 it would do 1 * 5, 2* 5, 3 * 5 and 4 * 5
 There are two errors in the code
 Identify each error and suggest a fix for both

```
function multiplier(n)
    for I = 1 to n
        int count = 1
        count = count * n
        next i
    return count
end function
```

Error in line 3 - the count variable is set to 1. Each time the loop repeats the count will be reset to 1. It needs to be declared outside the loop

*Error in line 4 - the count variable is multiplied by n but it should be multiplied by i. Line 4 should read count = count * i*

TEST PLAN

Type of Test Data	Test Data	Reason for Testing	Expected Outcome
Normal	2345	To see if an integer value can be entered	Number will be allowed
Extreme	0001	To see if the lowest possible value can be accepted	Number will be allowed
Erroneous	No Input	Input box should prompt for a number	Error message in screen