* 1. An error in the rules/grammar of the language with any suitable example.
	2. Error messages/translator diagnostics produced when translating/by the compiler or on the fly while writing code. Attempts to tell you what the error is and indicate where the error is/line numbers/underlines. Editor allows you to enter the corrected code.
	3. 0, 12, 4.
	4. Set of test data: 1 1 3 / 1 1 4 / 1 1 5 / 1 1 6 / 2 2 5 / 2 2 6
	Expected output: -1 -2 -3 -4 -1 -2
	5. A data structure/collection of several variables under one name. Each individual variable is given an index by which it is referred within the array.
	6. Data type: Integer. Reason: A dice roll is always a whole number between 1 and 6.
	Size: 3. Reason: one element is needed for each dice.
	7. Example (also accept similar with FOR-NEXT loop)

	**BEGIN RollTheDice
	 i = 1
	 WHILE i <= 3
	 DiceRoll(i) = Random No**

 **i = i + 1
 END WHILE
END**

* + 1. A name/symbol which represents a value is a program. The value can change while the program is running.
		2. ORIGIN
		String. Consists of more than one character.
		Size
		Integer. Consists of whole numbers.
	1. Dress A : 14
	Dress B : 10
	Dress C : 12
	2. Coins(4)=50, Coins(10)=0.
		1. The program is written to do something other than what the programmer intended
		2. It will only reset the first 9 elements / will not reset the 10th element. After setting Coins(9) = 0, i will become10 and the loop will stop. It should be UNTIL i > 10 / or other working correction
	3. Example:
	**i = 1
	total = 0
	REPEAT
	 total = total + Coins(i)
	 i = i + 1
	UNTIL i>10 or Coins(i)=0**
	**OR:**
	**total = 0
	FOR i = 1 to 10
	 total = total + Coins(i)
	NEXT i**

		1. Pi
		2. WheelSize & Circumference
		3. The value of a constant cannot be changed once the program is running (can only be set at design time). The value of a variable can change as the programming is running and has no value at design time.
	4. An integer is a whole number. A real number can include decimal fractions.
1. Example:
**BEGIN
 Input RealAge
 IF RealAge <= 2
 DogYears = RealAge \* 12
 ELSE
 ExtraYears = RealAge – 2
 DogYears = 24 + ExtraYears \* 6
 END IF
END**

	1. A name which is used to identify a memory location used to store a value which can change.
		1. A=4, B=9
		2. A=2, B=2
	2. Example:
	**If A > B Then
	 Temp = A
	 A = B
	 B = Temp
	End If**
	3. Message: String
	Number of flashes: Integer
	4. 
	5. Network: string/text/alphanumeric
	CallLength: real /float/single/double
	SameNetwork: Boolean
	TotalCalls: integer
	RunningCost :currency/real

		1. TotalCalls=11
		RunningCosts=12.01
		2. TotalCalls=12
		RunningCosts=12.51
	6. The keyword AND has been misspelled (ADN). This breaks the rules of the language and is a syntax error.
	7. It will produce the wrong result as it is adding instead of subtracting. This is a logic error.

		1. The value of WordList(6) is “**mama**”
		2. The value of WordList(**9**) is “taso”
	8. EXAMPLE:
	**INPUT SearchWord
	I = 0
	REPEAT
	 I = I + 1
	 IF WordList(I) = SearchWord THEN
	 OUTPUT “Word Found”
	 END IF
	UNTIL I = 10**
	9. High level languages (HLL) are understood by humans. Computers/the CPU can only understand/execute machine code instructions. The translator converts a program in the HLL to an equivalent program in machine code.
	10. A compiler translates the entire program before execution. An interpreter translates one line, executes that line and then translates the next line. A compiler creates a list of errors after compilation. An interpreter stops after the first error. A compiler produces an independent executable file. An interpreted program needs the interpreter each time it is run. A compiled program is translated once. An interpreted program is translated each time it is run.