Section 11.1 - Programming Basics

Layer 6: High-Order Language

Syllabus Content Section 11: Programming

S11.1.1 Implement and write pseudocode from a given design presented as either a program flowchart or structured English

---

S11.1.2 Write pseudocode statements for:

- the declaration of variables and constants
- the assignment of values to variables and constants
- expressions involving any of the arithmetic or logical operators input from the keyboard and output to the console

---

> Variable Declarations

```
DECLARE <identifier> : <data type>
```

- EXAMPLE - variable declarations

```
DECLARE Counter : INTEGER
DECLARE TotalToPay : REAL
DECLARE GameOver : BOOLEAN
```

> Constants declarations

```
CONSTANT <identifier> = <value>
```

- EXAMPLE - CONSTANT declarations

```
CONSTANT HourlyRate = 6.50
CONSTANT DefaultText = "N/A"
```

Assignments

> assignment operator: "←"

```
<identifier> ← <value>
```

- EXAMPLE - assignments

```
Counter ← 0
Counter ← Counter + 1
TotalToPay ← NumberOfHours * HourlyRate
```

Input and output

```
INPUT <identifier>
OUTPUT <value(s)>
```

- Example – INPUT and OUTPUT statements

```
INPUT Answer
OUTPUT Score
OUTPUT "You have ", Lives, " lives left"
```

Arithmetic operations

+	Addition	
-	Subtraction	
*	Multiplication	
/	Division	The resulting value should be of data type REAL, even if the operands are integers.)
DIV	Integer division	Used to find the quotient (integer number before the decimal point) after division
MOD	Modulus	The remainder that is left over when one number is divided by another.

Relational operations

>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<>	Not equal to

Logic operators

```
AND, OR, NOT
```

S11.1.3 Use built-in functions and library routines

- Any functions not given in the pseudocode guide will be provided
- String manipulation functions will always be given

`String and Character Functions`		
LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING	returns leftmost x characters from ThisString	Example: LEFT("ABCDEFGH", 3) returns "ABC"

`String and Character Functions`		
RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING	returns rightmost X characters from ThisString	Example: RIGHT("ABCDEFGH", 3) returns "FGH"
MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING	returns a string of length y starting at position x from ThisString	Example: MID("ABCDEFGH", 2, 3) returns string "BCD"
LENGTH(ThisString : STRING) RETURNS INTEGER	returns the integer value representing the length of string ThisString	Example: LENGTH("Happy Days") returns 10
LCASE(ThisChar : CHAR) RETURNS CHAR	returns the character representing the lower-case equivalent of ThisChar. Alphabetic characters that are not upper case are returned unchanged.	Example: LCASE('W') returns 'w'
UCASE(ThisChar : CHAR) RETURNS CHAR	returns the character representing the upper-case equivalent of ThisChar. Alphabetic characters that are not lower case are returned unchanged.	Example: UCASE('a') returns 'A'
TO_UPPER(ThisString : STRING) RETURNS STRING	returns a string formed by converting all characters of ThisString to upper case.	Example: TO_UPPER("Error 803") returns "ERROR 803"
TO_LOWER(ThisString : STRING) RETURNS STRING	returns a string formed by converting all characters of ThisString to lower case.	Example: TO_LOWER("JIM 803") returns "jim 803"
NUM_TO_STR(x : "datatype1") RETURNS "datatype2"	returns a string representation of a numeric value. Note: "datatype1" may be REAL or INTEGER, "datatype2" may be CHAR or STRING	Example: NUM_TO_STR(87.5) returns "87.5"
STR_TO_NUM(x : "datatype1") RETURNS "datatype2"	returns a numeric representation of a string. Note: "datatype1" may be CHAR or STRING, "datatype2" may be REAL or INTEGER	Example: STR_TO_NUM("23.45") returns 23.45
`String and Character Functions`		
IS_NUM(ThisString :) RETURNS BOOLEAN	returns the value TRUE if ThisString represents a valid numeric value. Note: may be CHAR or STRING	Example: IS_NUM("-12.36") returns TRUE
ASC(ThisChar : CHAR) RETURNS INTEGER	returns an integer value (the ASCII value) of character ThisChar	Example: ASC('A') returns 65, ASC('B') returns 66, etc.
CHR(x : INTEGER) RETURNS CHAR	returns the character whose integer value (the ASCII value) is x	Example: CHR(65) returns 'A', CHR(66) returns 'B', etc.
`Numeric Functions`		
INT(x : REAL) RETURNS INTEGER	returns the integer part of x	Example: INT(27.5415) returns 27
RAND(x : INTEGER) RETURNS REAL	returns a real number in the range 0 to x (not inclusive of x).	Example: RAND(87) could return 35.43
`Date Functions`	Date format is assumed to be DD/MM/YYYY unless otherwise stated.	

`String and Character Functions`		
DAY(ThisDate : DATE) RETURNS INTEGER	returns the current day number from ThisDate	Example: DAY(04/10/2003) returns 4
MONTH(ThisDate : DATE) RETURNS INTEGER	returns the current month number from ThisDate	Example: MONTH(04/10/2003) returns 10
YEAR(ThisDate : DATE) RETURNS INTEGER	returns the current year number from ThisDate	Example: YEAR(04/10/2003) returns 2003
DAYINDEX(ThisDate : DATE) RETURNS INTEGER	returns the current day index number from ThisDate where Sunday = 1, Monday = 2, etc.	Example: DAYINDEX(12/05/2020) returns 3
SETDATE(Day, Month, Year : INTEGER) RETURNS DATE	returns a variable of type DATE with the value of DD/MM/YYYY	
TODAY() RETURNS DATE	returns a variable of type DATE with the value set to the current date.	
`Text File Functions`		
EOF(FileName : STRING) RETURNS BOOLEAN	returns TRUE if there are no more lines to be read from file FileName	Note: The function will generate an error if the file is not already open in READ mode.