**Principles of Basic Language**

**Introduction:**

The program in basic language compound of **finite number** of **statements**. Each statement contains the following items :

1. Numbering ( the number of statement represent the recognized name of statement which differ one to another. This number must integer and prefer start from 10, 20,30,….).
2. Key word (such as READ, INPUT , IF, PRINT ,GOTO ,……etc. that represent a determine order).
3. Mathematical expression (such as equations for addition or multiplying or divisions or subtractions ….. etc)

**Example1 :**

**Numbering**

10 Input x

 10 Input x

**Key word**

**space**

 20 y = x+5

 30 Print y

 40 End

**Principles of Basic language :**

1. Principle symbols :

Consist of

1. Digits : (0,1,2,3,…………..9).
2. English letters : (A,B,C,D,…………………Z)
3. Special symbols :

|  |  |
| --- | --- |
| Symbols | Meaning |
| + | **Addition** |
| − | **Subtraction** |
| × | **Multiply** |
| / | **Division** |
| ^ or \*\* | **Power to number** |
| [ ] , ( ) | **bows** |
| = | **Equal** |
| > | **Greater than** |
| < | **Less than** |
| => or >= | **Greater than or equal** |
| =< or <= | **Less than or equal** |
| < > or > < | **Not equal** |
| , | **Comma** |
| ; | **Point comma** |
| “ “ | **double** **quotes** |

1. **Constants and Variables :**
2. **Constants** : are the values that remain constant during running the program and consist of :

a-1 **Numerical constants** : represent numbers and consist of :

**a-1-1 INTEGER CONSTANTS :** Whole numbers written without the decimal point such as (89,132,7698,+10,-8,2,2%=2)

 هي عبارة عن الاعدادالتي لاتحتوي على فاصلة عشرية او على أي جزء كسري وقد تحمل هذه الاعداد الاشارة السالبة او علامة النسبة المئوية التي تشير الى ان العدد الصحيح مقرب الى اقرب عدد صحيح

**a-1-2 Real Constants :** the numbers used the decimal form such as(6.9,-52.76,0.095, -3269.0)

هي الاعداد التي تحتوي على فاصلة عشرية او على جزء كسري وكذلك يلحق العدد بعلامة ! التي تشير الى الدقة الاحادية وعلامة تشير الى الدقة المضاعفة

**a-1-3 Exponential Constants :** this form requires a number fallowed by the letter E, such as(5E4= 5\*104, 12E-3=12\*10-3).

E وهي الثوابت التي تكون مرفوعة الى قوى اسية ويعبر عنها بالحرف

 **a-2 String Constants :** a string consists of a sequence of characters enclosed in double quote marks. Strings usually consist of names or address or cities such as "Computer", “1250”,”x+y”.

ثوابت السلسلة الحرفية وهذا النوع من الثوابت غير خاضع للعمليات الحسابية ويكون محصوربين علامتي المتن(" ") حيث تجري العمليات على ما موجود داخل هذه العلامات وعادة تستخدم لكتابة العناوين.

1. **Variables** : A variable is a **name** which can contain a **value**. Variables are items which may change their values during the execution of a program. Variables are spots in memory that hold information (we call that information DATA). Programming involves giving values to these variables and using the information stored in them in a useful way.

**Rules For Naming Variables In Basic:**

1. The first character must always be a letter(i.e A,G,D,F,z,e,t, or y)

2. After the very first letter in a variable name you can have letters, digits or underscores.

3. The final character MAY either be %,&,!,#,$ or nothing.

4. Remember variable name's cannot be reserved words. For example in QBasic there is a statement called "PRINT", you must not use this or any other statement name for a variable name.

5. A variable can be a maximum of 40 characters long.

6. In QBasic it does not matter if your variable names are either uppercase or lowercase. For example if you type a variable name as COST and then later in the program you type the variable name as cost, you will find that the variable name will match any other in the program.

**Variable Types:**

b-1 **Numerical Variables :** holds a number and are divided into the following types : Integer, Long Integer, Single Precision, Double Precision. Such as (A,b,c…..,A0,b1,c2,Ab,ba,bv,zx)

 b-2 **String Variables** : holds something other than a number. This variable contains values within the double quotes (" "). A $(dollar) sign has to be added to show the string variable such as (A$,BC$,A2$,ZW$).

3- **Types of Operation in Basic Language :**

3-1 **Arithmetic Operations:** Arithmetic Operators are used do some kind of calculation with numeric values. They cannot be used with strings. The Arithmetic Operators in BASIC are the as :

|  |  |  |
| --- | --- | --- |
| Operator | **Function** | **Usage** |
| + | addition | C = a + b |
| - | subtraction | C = a-b |
| \* | multiplication | C = a\* b |
| / | division | C = a/b |
| \*\* or ^ | Exponential | C = eb |

as well as that we can use the expression MOD to division the integer number and find the remain such as

10 MOD4 = 2

The order of precedence in performing arithmetic operations:

|  |  |  |
| --- | --- | --- |
| Symbol | Order | Remark |
| ( ) | first | Operations within parentheses are performed first. |
| ^ | second | Exponentiation is performed before other arithmetic operations. |
| \*, / | third | Multiplication and division are equal in order of precedence. |
| + , - | forth | Addition and subtraction are performed last and are equal in order of precedence |

 **Example1 :** Suppose( a=5. b=4 ,c=2 ,i=2,j=1) Execute the following expressions in order of precedence:

? a + b + c / 12

?(a + b+ c) / 12

? a + b / c – I \* j + a ^ 2

? a \* b – c / I + j + a ^ b + c ^ 3

? a \* j ^ b – I / c \* j + 20

? j / ( I + c )

? j / I +c

? ( a \* ( b + I ) \* c ) ^ 2

? ( j + I \* ( b / c – a ^ 2 ) ) + i

? ( I + j ) / a

? I + j / a

? ( j \* ( a ^ 2 – 10 ) ) / 2

? i + j ^ 2 – 3 \* a

? ( ( I + j ) / 3 \* a ) ^ 2 / 2

* 1. **Relational Operations :** Relational Operators are used to **compare** **two values.** They form relational expressions. These operators will always give a single value as result **True** or **False**. The Relational Operators in BASIC are the following as :

|  |  |
| --- | --- |
| Operator | Meaning |
| =  | Equal to |
| < > | Not equal to |
| > | Greater than |
| < | Less than |
| > = | Greater than or equal to |
| < = | Less than or equal to |

* 1. **Logical operations :** Logical Operators are used to combine **two or more relational expressions.** The Logical Operators in BASIC are the Following:

**NOT , AND , OR**

These operators also will always give a single value as result True or False.

**Functions in Basic Language :**

These functions are indicated by three letters naming the function, followed by argument enclosed in parentheses. The argument may be either a number, a variable, or a an expression. In the following table library functions as might be found in most basic interpreters.

|  |  |
| --- | --- |
| Function | Description of function |
| ABS(X) | Absolute value of x ,¦x¦. |
| PI | The value of π |
| SQR(X) | Square root of x. |
| INT(X) | The largest integer not exceeding x. |
| RND(X) | Create random number value between 0 and 1 |
| SIN(X) | Sin of x, x in radians |
| COS(X) | Cosine of x, x in radians |
| TAN(X) | Tangent of x, x in radians |
| ATAN(X) | Arctangent of x, x in radians |
| EXP(X) | The value of ex  |
| LOG(X) | The natural logarithm of x ,ln(x) |
| LGT(X) | The decimal logarithm of x ,log(x) |

**Example2 :** Convert following mathematical expressions in appropriate formula QBASIC language:

S=$sin(x)^{2}$+ $cos(x)^{3}$ →S=sin((x\*3.14/180)^2)+cos(x\*3.14/180)^3

D=log(x)-$ log(x)^{2}$ →D=log(x)/log(10)=log(x^2)/log(10)

R=¦d-s¦ / n →R=abs(d-s)/n

Y=1-ex /1+e x  →Y=(1-exp(x))/(1+exp(x))

Y=$s\sqrt{s}$+4 →Y=s\*sqr(s)+4

Y=logb-sinx+dr →Y=log(b)/log(10)-sin(x\*3.14/180)=d\*r

D=logx-b →D=log(x)/log(10)-b

V=logm-logk →V=log(m)/log(10)-log(k)/log(10)

B=lnx +bln y+b ln x2 →B=log(x)+b\*log(y)+b\*log(x^2)