

# COMPUTER PROGRAMMING IN QBASIC

## COMPUTER PROGRAMMING IN QBASIC

The full form of QBASIC is Quick Beginners All-purpose Symbolic Instruction Code. QBASIC is the most popular high-level programming language. Various versions of BASIC have been developed by Microsoft Company. This language is quite simple to understand and has been adopted by most of the microcomputers. It is suitable for both mathematical and business problems. It is compatible with MS-DOS environment and it has two basic files QBASIC.EXE and QBASIC.HLP. We can edit, debug and execute the program using these two files.

### Advantages of QBASIC

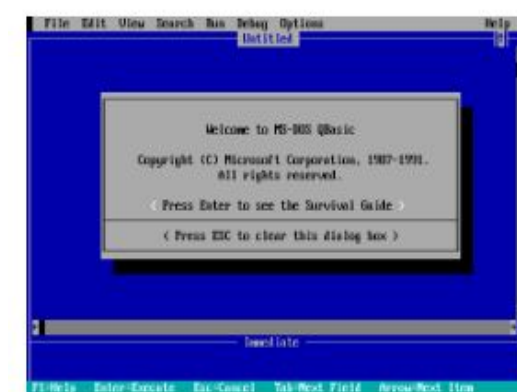
QBASIC is easy to learn and fun to practice program. It may be called a "People's language".

It is available almost for every computer from micro-computer to mainframe. Therefore, a program developed in a micro-computer can run on a bigger system with minor modifications.

It is suitable for mathematical and business application.

Program development cycle is quick, debugging is simple, and

Modification of the program is quite easy.



### Starting QBASIC(from DOS)

Turn on the computer.

Click on start button and get DOS prompt.

Change the directory to QBASIC.

C:\> CD QBASIC [Press Enter Key]

C:\> QBASIC > QB [Press Enter Key]

Now you will see the screen:

Ac  
Go

# ELEMENT OF QBASIC



To construct the QBASIC program, we have to arrange some standard elements. The elements used in Qbasic are as follows:

- Character set
- Variables
- Constants
- Operator and Operands
- Expression
- Statements

## Character Set

QBASIC has the character set consisting of the following elements:

1. Alphabets: A, B, C, ..., Z
2. Digits: 0, 1, 2, ..., 9 and
3. Special characters: +, -, \*, /, (, ), ., \$, :, ;, =, >, <, ^

The symbol ^ (caret) is used to denote exponentiation operator, the symbol \* (asterisk) is used to denote multiplication and other symbols; have their usual meanings.

## Constants

A quantity in a computer program which does not change its value during the execution of the program is called a constant. QBASIC allows the following constants:

### Numeric constant

The numeric constant is one that is formed by a sequence of digits 0, 1, 2, ..., 9 and may include a decimal point. A numeric constant may be an integer or a real number. 383, +57, 0, -6.2 and 6.15E4 are valid numeric constants. The number 6.15E4, in fact, represent 6.15 \* 10<sup>4</sup>. The notation E is used to represent the exponential form. The number after E is the exponent which can be positive or negative. However, its length cannot exceed two digits.

It is also important to keep in mind that:

- QBASIC does not distinguish between an integer and fraction.
- Commands are not allowed in a numeric constant.

The limit on the number of digits that can be used varies from computer to computer. Normally, a numeric constant can have up to a maximum of eight digits.

### String constant

A string constant consists of a sequence of characters which must be enclosed by a quotation mark.

## Variables

The quantity which may change its values during the execution of the program is called the variable.

In QBASIC, variables are also of two types:

**Numeric variable:** Numeric variable can assume numeric value and is represented by an alphabet or an alphabet followed by another alphabet or digit. For example A, C, A2, ABC, A6 etc, represent numeric variables.

**String variable:** A string variable is represented by an alphabet followed by dollar (\$) sign. It should be kept in mind that while constructing the string variable, dollar (\$) should be the last character. For example, B1\$, NAME\$, BOOK1\$, etc are valid string variables.

## Expression

An expression can be a string, or numeric constant, a variable or a combination of constants, variables with operators which returns a single value.

## Operands

Operands are the data or variables on which mathematical, logical and string operations take place.

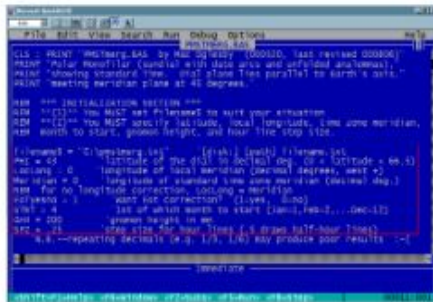
## Operators

Operators are the symbols, which are used in arithmetic operations, logical expressions, and string expressions.

## Statements

A statement is a set of instructions written using keywords or commands of QBASIC. Every programming language uses keywords as a statement with certain syntax.

## STATEMENTS IN QBASIC



A statement (for the QBASIC) is a set of instructions written by using keywords or commands of QBASIC. Every programming language uses keywords as a statement with certain syntax. The keywords have specific meaning in the QBASIC programming. The statements are the first stored in the memory and executed only when the RUN command is given.

Different statements used in QBA SIC are as follows:

### CLS Statement

The CLS statement clears the screen. If you write CLS statement in the middle of the program then you cannot see the outputs generated before execution of CLS because it clears the screen.

**Syntax:** CLS

### LET Statement

LET is an assignment statement. It is used to assign the value to a variable. LET is an optional statement i.e. without using LET statement one can assign the value to a variable. The data type must match with the variable type otherwise type mismatch error will occur.

Syntax: **LET** variable = value or expression

Example:

CLS

INPUT " First Number"; A

```
INPUT "Second Number": B
```

```
let Sum = A+B
```

```
PRINT " The Sum is"; S
```

END



## REM Statement

It is a basic declaration statement that allows explanatory remarks to be inserted in a program. The remarks may be useful in a program to explain about different kinds of statements and user defined words. Adding comments in the program allows us to remind about the program and also helps other programmers to understand the logic of the program.

**Syntax:** REM < Remarks>

### Example:

```
CLS
PRINT "Some text."
REM This text is ignored.
REM This program clears the output screen.
REM and then PRINT " Some text."
```

## PRINT Statement

PRINT statement provides output on the screen. It prints the values of the expression on the screen. If the expression list is blank, no characters are printed. The expressions in the list may be numeric or string. In case of number, the negative number is preceded by a minus sign (-) but in positive number it is preceded by a space.

We can use semicolon and comma with a print statement which results differently than a normal PRINT statement. If expression list ends with comma or semicolon, the next PRINT statement prints on the same line. Comma provides a TAB space, but semicolon provides only one space.

**Syntax:** PRINT ["Message"]; expression

### Example:

```
CLS
PRINT "Computer is an electronic machine."
PRINT
PRINT "IT's amazing."
PRINT 1000
PRINT " The number is: "; 20
END
```

### Output

```
Computer is electronic machine
IT's amazing
1000
The number is: 20
```

Ac  
Go