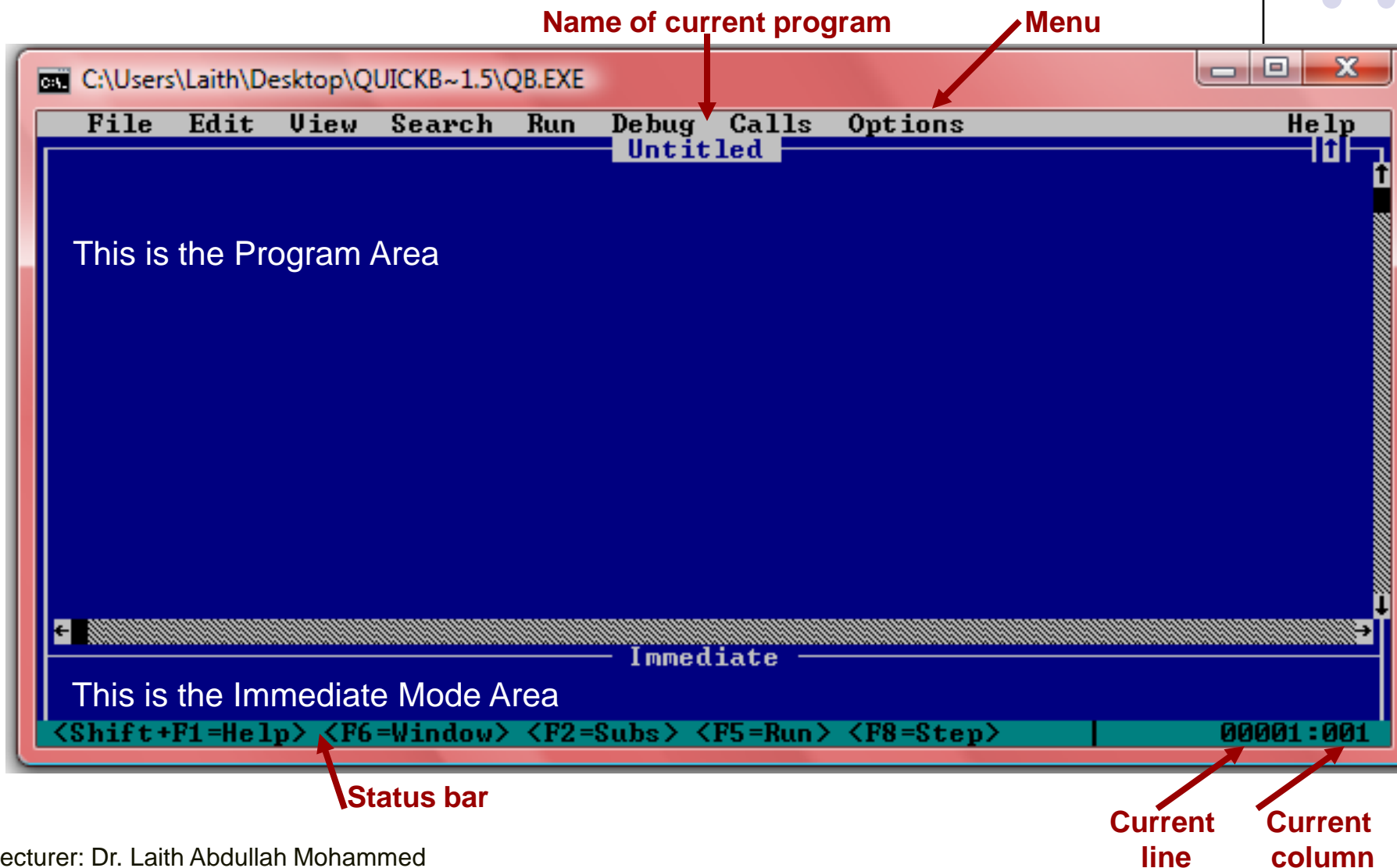
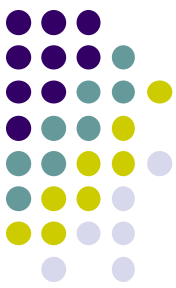




QBasic stands for **Beginner's All-Purpose Symbolic Instruction**. It is a programming language written for computers back in 1975, by Bill Gates & Paul Allen. It is ease of use , its English-like commands and its power.





●The Character set:

1. The Letters: its from A to Z (small or capital).
 2. The Digits: its from 0 to 9
 3. Special symbols: like (+,-,^,?,!,#, etc.)
-

●The Constants:

1. Numeric constants: include all numbers (real, not real, integer,).
Example: 25, -230, 0, 16.44, 0.88
 2. Character constants: include all characters sets (letters, digits, symbols) between two “ “. Example: “ BASIC
“, “ The width is = 83 “, “ Telephone Number 07901 “.
-

●The Variables:

A variable is a name which can contain a value. The variables must include the conditions below:

1. From A to Z (include all letters).
2. Not contained symbols except dot (.).
3. Maximum length of variable is 40.
4. Must not represent any word which is defined as a special word in QBASIC.
5. Must be start by letters.

A. Numeric Variables:

Like: M, A2, WE,.....etc

In Numeric variables the symbol (%) (**Integer**) mean make the numeric variable as real number.

Like: A%=6.2	its mean A%=6
A%=6.6	its mean A%=7
A%=6.5	its mean A%=7
A%=-6.5	its mean A%=-6

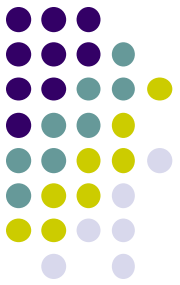
Using symbol (&) (**Long**) with numeric variable make it long variable.

Using symbol (!) (**Single**) with numeric variable mean the length of variable equal to 7 digits or less.

Using symbol (#) (**Double**) with numeric variable mean the length of variable more than 7 digits.

B. Character (String) Variables: If the variable holds symbols or text, it may be a character variable or a string variable. like: M\$, A2\$, WE\$,.....etc

■ Numeric variables:



Single-precision variables (floating-point variables): these types of variables are used to store numbers that contain a decimal value such as 1.89 or 3.141593.

INTEGER: A non-floating point variable (no decimal value) that can store integers between -32,768 and 32,767

LONG: same as INTEGER, but can contain numbers between -2,147,483,648 and 2,147,483,647

DOUBLE: same as SINGLE, but can have twice as many digits. (like: 983288.18)

SINGLE: single precision variables. (like: 39.2932)

To define a variable's type, use **DIM** with the AS attribute.

```
DIM var1 AS INTEGER
DIM var2 AS LONG
DIM var3 AS DOUBLE
var1=15.28
var2=-2000000000
var3=12345678.12345678
PRINT var1
PRINT var2
PRINT var3
```

Output:

```
15
-2000000000
12345678.12345678
```

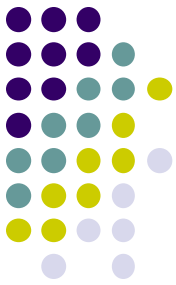
Example:

```
B=26
A$=" I Like to Learn QBASIC "
PRINT B
PRINT A$
```

Output:

```
26
I Like to Learn QBASIC
```

■ Strings variables:



String variables are ones that can hold all ASCII characters (Letters, Numbers, Symbols). They can not be used in math problems. When asking questions about them, or changing their content, the expressions must be in quotes. String variables are letters and numbers followed by a dollar sign (\$). String names must have a letter as the first character, but everything else is up to you.

Examples:

```
" 0123456789 "  
" abc123 "  
" 1+1=2 "  
"!@&%$§?><°^ "  
" Hi "
```

Example:

```
X$= "Hello World ! "  
D$= "Hi "+X$  
PRINT X$  
PRINT D$
```

Output:

```
Hello World !  
Hi Hello World !
```

■ Expressions:

Operator	Function	Example	Result
+	Add	8+2	10
*	Multiply	8*2	16
/	Divide	8/2	4
-	Subtract	8-2	6
^	Exponentiation	8^2	64

Example:

```
Z= 500+(10*7)  
rate=50  
time=2  
distance=rate*time  
PRINT Z  
PRINT distance
```

Priority of operations:

1. The parenthesis (from left to right & from inside to outside).
2. The exponentiation (from left to right).
3. Division & Multiply (from left to right).
4. Add & Subtract (from left to right).

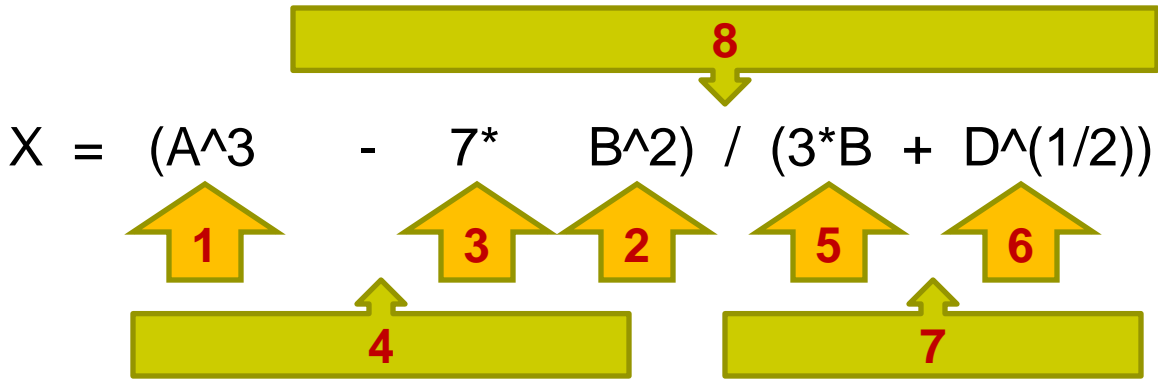
Output:

```
570  
100
```

Example: Write the following expression using BASIC format. And define the priority of the calculation for the expression

$$x = \frac{a^3 - 7b^2}{3b + \sqrt{d}}$$

Solution:

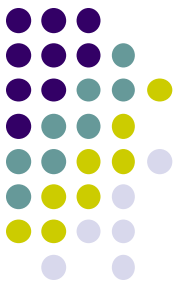


Example: Based on the priority of the calculation calculate the following:

2+3/4*5	⇒	= 5.75
2+3^2	⇒	2+9 = 11
3*4/5	⇒	12/5 = 2.4
3+4-5	⇒	7-5 = 2



Comparison Operators:



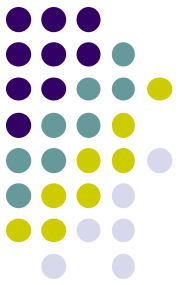
Operator	Meaning	Example
=	Equal to	IF a=15 THEN...
<>	Not equal to	IF a<>15 THEN...
<	Less than	IF a<15 THEN...
<=	Less or equal to	IF a<=15 THEN...
>	More than	IF a>15 THEN...
>=	More or equal to	IF a>=15 THEN...

Logical Operators:

1. **OR operator**: the result of this operator is true if one of the statements is true.

A	B	A OR B	Example
True	True	True	IF 10<14 OR 3^2=9 THEN...
True	False	True	IF 10<14 OR 3^2=5 THEN...
False	True	True	IF 10<2 OR 3^2=9 THEN...
False	False	False	IF 10<2 OR 3^2=5 THEN...

2. AND operator: the result of this operator is true when both statements are true.



A	B	A AND B	Example
True	True	True	IF $10 < 14$ AND $3^2 = 9$ THEN...
True	False	False	IF $10 < 14$ AND $3^2 = 5$ THEN...
False	True	False	IF $10 < 2$ AND $3^2 = 9$ THEN...
False	False	False	IF $10 < 2$ AND $3^2 = 5$ THEN...

3. NOT operator:

Example:

NOT $15 > 10$

NOT $15 = 8$

NOT $(5 \geq 5 \text{ AND } 6/2 = 4)$

False

True

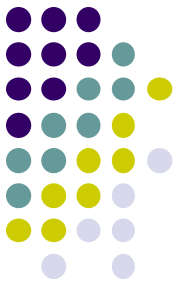
True

Note: ♦ The priority in Logical operations are:

1. The parenthesis ().
2. NOT operator.
3. AND operator.
4. OR operator.

♦ The priority between Mathematical Expression and Logical operations are:

1. Mathematical Expressions.
2. Logical Operations.



Example: Decide which statement is True and which is False.

A=3 , B=10 , C=100

According to Priority Law,

C=100

True

C<100

False

A=3 OR C>100 AND B<10

True

(A=3 OR C>100) AND B<10

False

B<=10 AND C=10 AND A*B=6 OR A<>13

True

B<=10 AND C=10 AND (A*B=6 OR A<>13)

False

Adding Documentation to the program:



Documenting your program allows you to remind yourself about something in your program. Plus, if your program is seen by other people, documenting can help them understand your code. The **REM** (remark) command enables you to add comments to your program without the text being treated like an instruction.

General Form:

REM or ' followed by comments.

Example:

```
CLS      ' This command clears the screen
REM Program to compute area
```

❑ **LET command:**

General Form:

LET variable=expression [Note: the word LET is optional].

Purpose:

Algebraic or string depending upon the variable – uses + - ^ \ MOD.

Example:

```
LET X= A+B*C
LET J=J+1
LET C=SQRT (A*A+B^2)
```