

COMPUTER PROGRAMMING

LECTURE 1 INTRODUCTION TO QBASIC

**DR. USMAN AKMAL
DR. SANA AMIR
ENGR. KHAWAJA AL**

**DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF ENGINEERING AND TECHNOLOGY,
LAHORE**



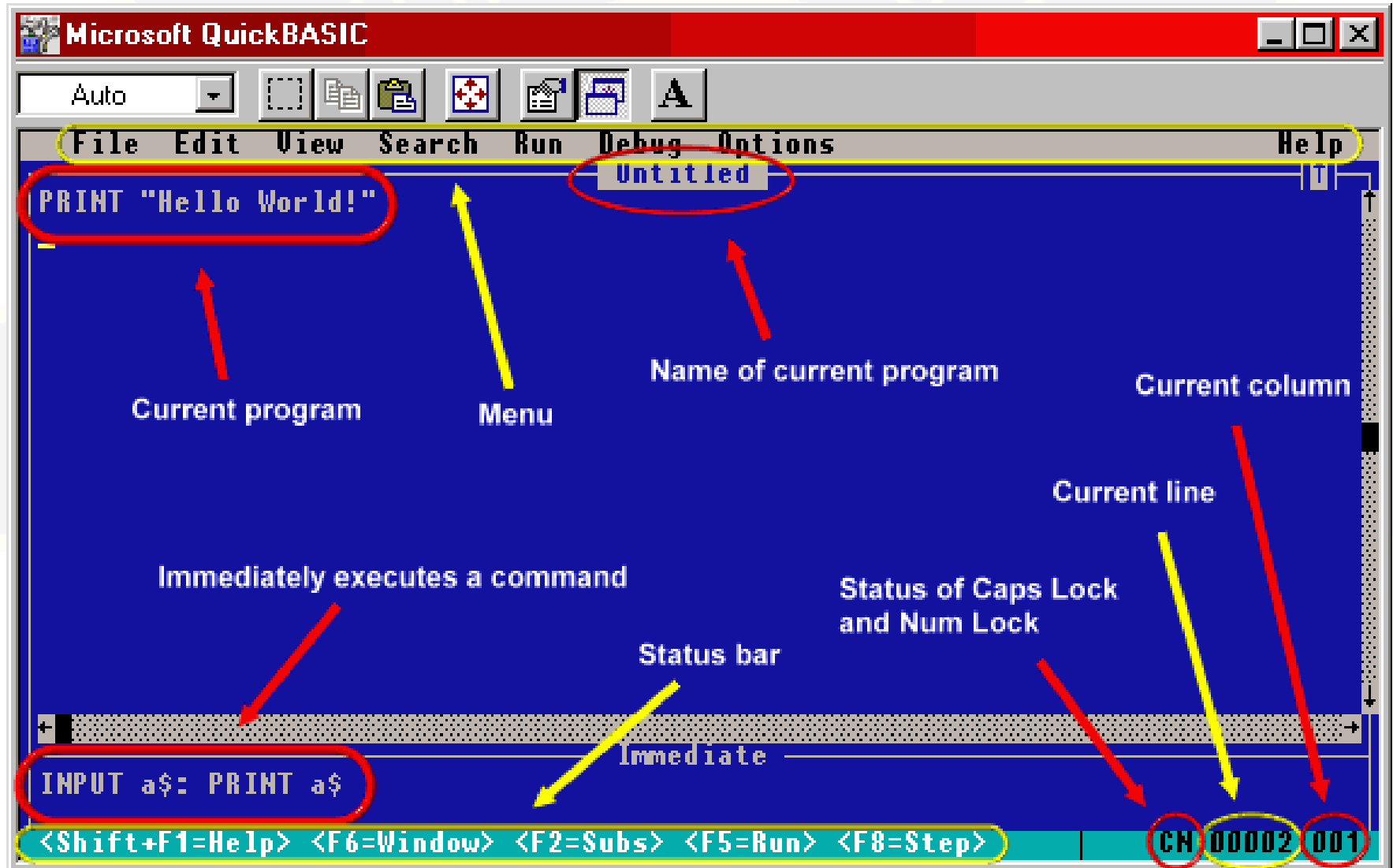
CONTENTS

- ❑ Introduction to QBasic Interface
- ❑ Data Types
- ❑ Variables
- ❑ **CLS** Command
- ❑ **INPUT** Statement
- ❑ **PRINT** Statement
- ❑ **END** Command

Introduction to QBasic Interface



Introduction to QBasic Interface



Introduction to QBasic Interface

1. Current Program / Coding Area

The current program / coding area is displayed in the middle of the screen, and covers most of the QBasic interface.

2. Menu Bar

The menu provides most of the operations for the QBasic editor. Such as opening a file, pasting text, and searching for a string.

Introduction to QBasic Interface

File Menu

New - Clears the current program

Open - Loads a program from disk

Save - Saves the current program to disk

Save As - Saves the program, but under a different name

Print - Prints the selected text, current window, or entire program

Exit - Closes the QBasic interpreter



Introduction to QBasic Interface

Edit Menu

Cut - Removes the selected text and stores it in the clipboard

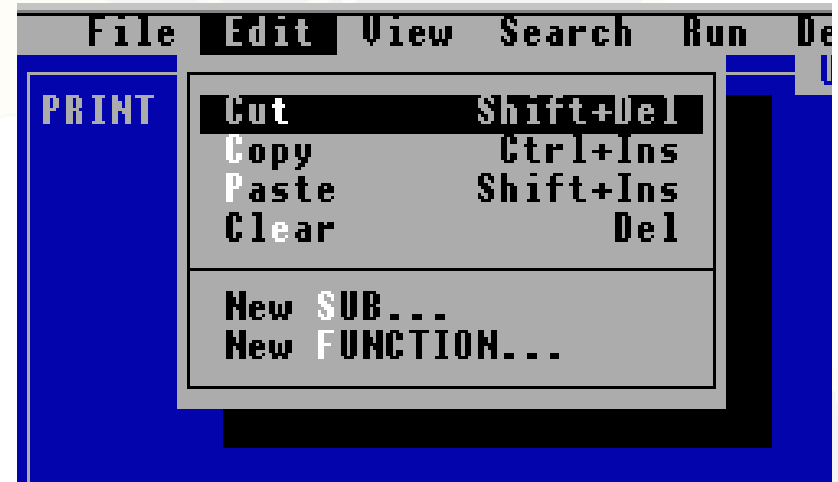
Copy - Copies the text instead of removing it.

Paste - Adds the text in the clipboard to the current position of the cursor

Clear - Removes the text without storing it to the clipboard

New Sub - Enables you to create a new subroutine

New Function - Enables you to create a new function



Introduction to QBasic Interface

View Menu

SUBs - Shows the list of current subroutines and functions

Split - Displays the contents of the current program in two windows. If the window is already split, this hides the second window

(NOTE: The text in each window is always the same, even if you alter the text in one window)

Output Screen - Shows the QBasic output screen.



Introduction to QBasic Interface

Search Menu

Find - Allows you to search for a string of text in the program

Repeat Last Find - Continues the previous search operation

Change - Replaces each instance of a string with another string



Run Menu

Start - Executes the current program

Restart - Starts from the beginning

Continue - Continues execution at the current position



Introduction to QBasic Interface

Debug Menu

Step - Processes the next command

Procedure Step - Processes the next command, but does not show QBasic going inside a subroutine or function

Trace On - Shows the command that is being executed while the program is running

Toggle Breakpoint - Sets or removes a breakpoint. Use this to have the QBasic interpreter stop when it reaches a specified line in the program

Clear All Breakpoints - Removes all breakpoints in the program

Set Next Statement - Allows you to continue execution at the specified line



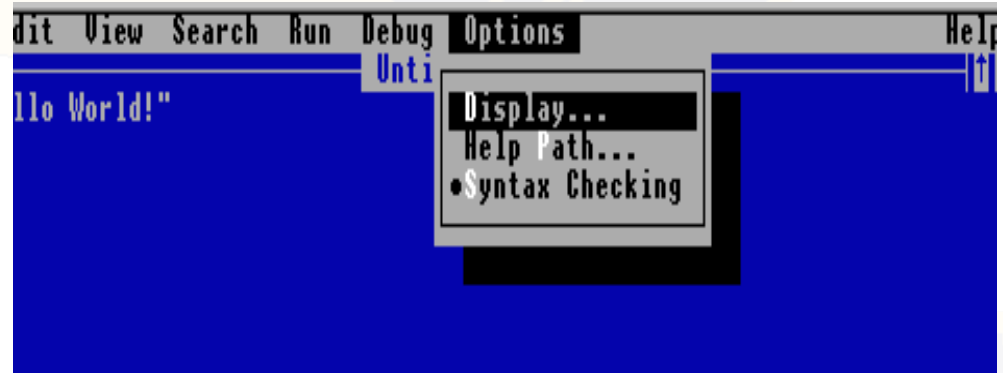
Introduction to QBasic Interface

Options Menu

Display - Enables you to change display colors, the number of spaces to use for tabs, and whether or not scroll bars are visible

Help Path - The location of the QBASIC.HLP file

Syntax Checking - Allows you to have the QBasic editor check the syntax of your program as you type



Introduction to QBasic Interface

Help Menu

Index - List of all QBasic commands, keywords, operators, etc.

Contents - The table of contents for QBasic help

Topic - Show help for a specific keyword

Using Help - Displays information on using QBasic help

About - Shows information about the QBasic editor



Introduction to QBasic Interface

3. Name of current program

The file name of the current program is displayed near the top of the screen in the center. You can change the name by selecting **"Save As"** on the **"File"** menu.

4. Immediately execute a command

QBasic provides a way to execute a command without running the current program. To do so, select the bottom window (under **"immediate"**) and enter a command, then press **Enter**.

Introduction to QBasic Interface

5. Status bar

The status bar is at the bottom of the screen. It displays a short list commands

<Shift+F1=Help>

<F6=Window>

<F2=Subs>

<F5=Run>

<F8=Step>

When you highlight an item on the menu, the status bar displays a short description of what the item does.

Introduction to QBasic Interface

6. Status of Caps Lock and Num Lock

If **Caps Lock** is set, a "C" is displayed on the right side of the status bar.

If **Num Lock** is set, a "N" is displayed on the right side of the status bar.

Introduction to QBasic Interface

7. Current line

On the right side of the status bar, the current line of the cursor is displayed.

8. Current column

On the right side of the status bar, the current column of the cursor is displayed (immediately after the current line).

DATA TYPES

DATA TYPES IN VB (QB)

BOOLEAN

BYTE

CURRENCY

DATE

INTEGER

LONG

SINGLE

DOUBLE

STRING

OBJECT

USER DEFINED

VARIANT

DATA TYPES (*contd.*)

Data Type	Memory Size (Bytes)	Range of Values	
		+ve	-ve
Integer	2	+32,767	-32,768
Long	4	+2,147,483,647	-2,147,483,648
Single	4	3.402823×10^{38} To 2.802597×10^{-45}	$-2.802597 \times 10^{-45}$ To -3.402823×10^{38}
Double	8	$1.79769313486231 \times 10^{308}$ To $4.940656458412465 \times 10^{-324}$	$-4.940656458412465 \times 10^{-324}$ To $-1.79769313486231 \times 10^{308}$
String	2+LEN	32,767	

VARIABLES

- Variables are names used to represent values that are used in BASIC Program.
- There are two types: **Numeric and String**
- A numeric variable has a value that is a number.
- A string variable may have a single character or many characters in it.
- A variable is a name that refers to an object--a particular number, string, or record. (A record is a variable declared to be a user-defined type.)

Variable NAMES

- A BASIC variable name may contain up to 40 characters.
- The characters allowed in a variable name are letters, numbers, the period (.), and the type-declaration characters (% , & , ! , # , and \$).
- The first character in a variable name must be a letter.
- A variable name can not contain space.

Variable NAMES

- A variable name cannot be a reserved word, but embedded reserved words are allowed.
- For example, `Log = 8` is illegal because `LOG` is a reserved word [BASIC is not case sensitive, i.e. `Sales`, `SALES` and `sales` all refer to the same variable]
- However, the following statement is legal;
`TimeLog = 8`
- Reserved words include all BASIC commands, statements, function names, and operator names.

VARIABLE DECLARATION

Simple variables can be numeric, string, or record variables. You may specify simple variable types in three different ways:

- i. AS declaration statement
- ii. Type-declaration suffix
- iii. DEFtype declaration statement

AS declaration TYPE

Defining the variable in a declaration type has the following form

declare **variablename** **AS** **type**

where the "declare" can be either **DIM**, **COMMON**, **REDIM** (for arrays), **SHARED**, or **STATIC**

and the "type" can be either **INTEGER**, **LONG**, **SINGLE**, **DOUBLE**, **STRING**, or a user-defined type.

AS declaration TYPE (*contd.*)

For example, the following statement declares the *variable a* as having a long-integer type:

```
DIM a AS LONG
```

More Examples

```
DIM X AS SINGLE, Y AS DOUBLE
```

```
DIM Xy AS STRING, I AS Long, j AS Byte
```

```
DIM k AS single, kk AS single
```

```
DIM XYZ AS STRING*30
```


AS declaration TYPE (*contd.*)

String variables declared in an AS STRING clause can be either variable-length strings or fixed-length strings.

Variable-length strings are expandable: their length depends on the length of any string assigned to them.

Fixed-length strings have a constant length, specified by adding “**number*” to the AS STRING clause, where number is the length of the string in bytes.

Type-declaration suffix

Append one of the type-declaration suffixes to the variable name as given in the following Table.

Suffix	Data Type
%	Integer
&	Long integer
!	Single-precision
#	Double-precision
\$	String

Type-declaration suffix (*contd.*)

You can assign a string constant to the variable of up to 32,767 characters, as in the example below.

```
A$ = "SALES REPORT"
```

Single precision is the default for variables without a type suffix.

Type-declaration suffix (*contd.*)

Examples

A% is integer type name

A\$ is string type name

A& is long integer type name

A! or **A** is single-precision type name

A# is double-precision type name

CLS Command

When a program reaches a line containing **CLS**, it erases the output screen. Following is the syntax of this command;

Syntax: **CLS**

INPUT Statement

A device I/O statement that reads input from the keyboard during program execution and stores it into a list of variables.

Syntax:

```
INPUT[ ; ]["PromptString" { ; | , } ]VariableList
```

Argument	Description
;	A semicolon immediately after INPUT keeps the cursor on the same line after the user presses ENTER.
PromptString	A string constant printed before the prompt character.
;	Prints a question mark at the end of the PromptString.
,	Prints the PromptString without a question mark.
VariableList	A list of variables, separated by commas, to accept the input values.

INPUT Statement (*contd.*)

- The INPUT statement causes the program to pause and wait for data. One may enter the required data at the keyboard.
- The entered data is assigned to the variables in variablelist.
- The number of data items that one supply must be the same as the number and type of variables in the list.

PRINT Statement

A device I/O statement that outputs data on the screen.

Syntax:

PRINT [expressionlist][{, : ;}]

- If expressionlist is omitted, a blank line is printed.
- If expressionlist is included, the values of the expressions are printed on the screen.
- The expressions in the list may be numeric or string expressions. (string literals must be enclosed in quotation marks.)

END Command

- One may use END command to the end of Qbasic programs.
- Although using END was required in earlier versions of BASIC, this command is now optional.
- Some people always use and END statement to eliminate any ambiguity on the part of readers as to whether they have reached the true end of the program.

Syntax: **END**

The background of the slide features a large, light blue watermark of the University of Engineering and Technology logo. The logo is circular with a gear-like outer edge. Inside the circle, there is a yellow banner at the top with the text 'UNIVERSITY OF ENGINEERING AND TECHNOLOGY' in blue. Below the banner is a white open book icon. At the bottom of the circle, there are three wavy yellow lines and a banner with the text 'LAHORE' in blue.

END OF LECTURE 1