

## Basic Programming Lesson 2

Name: \_\_\_\_\_ Score \_\_\_\_\_

### **String Constants**

A string constant is any list of characters enclosed in quotation marks. BASIC does distinguish between capital and lowercase letters in this case.

Ex. "first" "Hello"

To include a quote mark within a string constant, you have to type it twice.

**Sample Program :** Type the following and run

Print "He said ""I don't believe it!"" and grinned."

End

(Did your words run together? Why? Fix program if this happens.)

**Your Program:** Write a program below that prints to the screen ....

She said "You look great!" and grinned. Type it into computer and run it.

---

---

### **String Variables**

A string variable differs from a numeric variable by ending in a dollar sign (\$). The dollar sign counts as one of the 31 allowed characters. Ex. name\$, first\_name\$,

You can build complex string expressions by using concatenation and substrings:

**Concatenation:** The ampersand (&) sign is used for concatenation, or gluing together strings.

**Sample Program:** Type the following and then run the program.

Rem Concatenation

Let A\$ = "You "

Let B\$ = " are "

Let C\$ = " a "

Let D\$ = " great "

Let E\$ = " mathematician."

Rem Putting together words to make a sentence

Print A\$ & B\$ & C\$ & D\$ & E\$

End

(Note the spacing in the lines above. This is done so the words do not run together when you do the concatenation.)

Rearrange the concatenation line so that the sentence makes no sense. Run it. Write below your new concatenation line.

---

Key Commands:

**INPUT** Command:

The command **INPUT** allows you to enter information while you run a program. A question mark is displayed on the screen and the program pauses until you type in the information requested. You have to press **RETURN** for the program to continue.

The **INPUT** command is always followed by a variable or variables (numeric or string) to hold the information typed in.

Type the following programs and run them. Explore what they do.

Example:

```
Print "How old are you?"
Input age                      (age is a variable holding person's age)
Print                          (Print by itself gives a blank line)
Print "So you are "; age; " years old!" (semicolon is used between "" and variables)
End
```

Example:

```
Print "Say something."
Input A$
Print
Print "Did you say ... "; A$
End
```

If more than one item is requested, separate each item typed in by a comma.

Example:

```
Print "Enter two numbers."
Input x,y
Let sum = x + y
Print "The sum of "; x ; " and "; y ; " is "; sum
End
```

Example:

```
Print "Enter your name, phone number, and address."
Input name$, phone$, address$
Print
Print name$
Print phone$
Print address$
End
```

**Note:** If you enter the wrong type of information requested, the computer will ask you to retype the information. If not enough information is given, the computer will ask for more. If too much information is given, the computer will ignore it.

**LINE INPUT** command:

You can combine the **PRINT** prompt and the **INPUT** command into one line.

Example:

```
Line input "Enter a numerator, then a denominator. "; n, d
Print
Let quotient = n / d
Print n; " / "; d; " = " ; quotient
End
```

### **CREATING A PAUSE IN YOUR PROGRAM**

You can use the Get statement with a string variable to allow your program to pause during the running of it.

Example:

```
Print "Think of a joke to tell me."
Print "Press return when ready to continue."
Get A$
Print "What was your joke?"
Input joke$
Print
Print joke$; " Ha! Ha!"
End
```



