CE 150 COMPUTER PROGRAMMING

ASCII & UNICODE

by

ENGR. IRFAN-UL-HASSAN

1

ASCII & EBCDIC

- ASCII (pronounced as Skey) stands for AMERICAN Standard Code For Information Interchange.
- This was an alternative notation to EBCDIC that is Extended Binary Coded Decimal Interchange Code.
- A coding scheme developed by IBM for use with its computers as a standard method of assigning binary (numeric) values to alphabetic, numeric, punctuation, and transmission-control characters.

2

ASCII

- This is an 8 bit binary notation of the decimal numbers 0-9, alphabetic a-z and A-Z , and other symbols like punctuation (‘ ; : . ,), arithmetic operators (+ - * /) and many other special characters and control codes.
- In 8 bit a total of 256 symbols can be defined.

3

ASCII Codes for 0-9

CHAR	DEC	ASCII	CHAR	DEC	ASCII
0	48	0011 0000	5	53	0011 0101
1	49	0011 0001	6	54	0011 0110
2	50	0011 0010	7	55	0011 0111
3	51	0011 0011	8	56	0011 1000
4	52	0011 0100	9	57	0011 1001

4

ASCII Codes for A-J

CHAR	DEC	ASCII	CHAR	DEC	ASCII
A	65	0100 0001	F	70	0100 0110
B	66	0100 0010	G	71	0100 0111
C	67	0100 0011	H	72	0100 1000
D	68	0100 0100	I	73	0100 1001
E	69	0100 0101	J	74	0100 1010

5

Conversion CHAR to DEC

- Quick Basic and Visual Basic have a function which gives DECIMAL Value equivalent to ASCII code of a CHARCTER.
- ASC is a string processing function which returns a decimal numeric value that is the ASCII code for the first character in a string.

6

Conversion CHAR to DEC

- ASC("A") ⇒ 65
- ASC("a") ⇒ 97
- ASC("0") ⇒ 48
- ASC("Z") ⇒ 90
- ASC("ABcX01") ⇒ 65

7

Conversion DEC to CHAR

- **Opposite of ASC is CHR$ in Quick Basic and CHR in Visual Basic.**
- **CHR$(Number) where number ranges 0-255**
- **It is a string processing function that returns a one character string whose ASCII code is the Number provided as argument.**

8

Conversion DEC to CHAR

- **CHR$(65) ⇒ “A”**
- **CHR$(97) ⇒ “a”**
- **CHR$(48) ⇒ “0”**
- **CHR$(90) ⇒ “Z”**
- **CHR$(256) will generate an ERROR.**

9

Limitations of ASCII

- **The 8-bit ASCII notation is sufficient for many practical applications in ENGLISH language or other languages having a smaller number of alphabetic.**
- **However, there are languages which have more than 256 alphabetic, typical examples are Far Eastern languages like CHINEESE, JAPANEESE, and KOREAN. (Thousands of Characters)**

10

UNICODE

- **If a computer application is required for more than one language, the total number of alphabetic required would again be very large. That can not be accommodated in 8 bits.**
- **For that purpose a 2B or 16 b code has been devised which is termed as UNICODE.**
- **It may contain 2^{16} = 65536 patterns. That can interpret alphabetic of any language or in fact of many languages.**

11

Data Type STRING

- **The string data is stored by ASCII notation using 1B per character. The longest string may consist of approximately 32767(65536) Bytes . The length of string is noted in 2B true valued whole number in QB.**
- **First of all the length of the string is coded in 2B as true valued whole number and then the ASCII codes for various characters of the string are coded in order.**
- **Thus the sentence which is 8 characters long will be noted in 10 Bytes.**

12

STRING to ASCII

- **The string "He is OK" will be coded as:**
- **00000000 00001000 01001000 01100101 <u>00100000</u> 01101001 01110011 <u>00100000</u> 01001111 01001011**

13

ASCII to STRING

- **The string obtained from the following sequence in memory**

  00000000 00001001 01010111 01100101 01101100 01101100 <u>00100000</u> 01100100 01101111 01101110 01100101

  is "Well done".

  The length of the string is 9.

14

A Program in QBASIC

```
For I=1 to 26
      print CHR$(I+64), CHR$(I+96)
Next
```

These three lines of code will print A a to Z z on the computer screen (Monitor)

15

CURRENCY data type

- **It is a data type in VB only.**
- **This data type stores floating-point numbers with four decimal places using 64 bits or 8B.**
- **The currency data type can represent numbers in the range -922 337 203 685 477.5808 to 922.337 203 685 477.5807**

16

DATE data type

- **It is also a data type in VB.**
- **The date type DATE takes 8B for storage.**
- **It can give any date between 1.1.100 to 31.12.9999.**
- **The date data type contains information about TIME also.**

17

BOOLEAN data type

- **It is also a data type in VB only.**
- **It takes one of the two values TRUE or FALSE.**
- **A zero means false. Any non-zero is true.**
- **Truly speaking –1 is true.**
- **Although 1bit or maximum 1B was sufficient to store a BOOLEAN, however, 2B are used for efficiency.**

18

Other data types in VB

- There are three more data types supported by VB.
- Those are Variant, Object and User- Defined.
- It is not proper to discuss these types at this stage .
- USER-DEFINED is supported by QB also.

19

DATA TYPES IN VB (QB)	
BOOLEAN	BYTE
CURRENCY	DATE
DOUBLE	INTEGER
LONG	SINGLE
STRING	OBJECT
USER DEFINED	VARIANT

20

CONSTANTS

- CONSTANTS are predefined values that do not change value during execution.
- There are two type: String (or Character) constants and Numeric constants

21

String CONSTANTS

- A string constant is a sequence of alphanumeric characters enclosed by double quotation marks (" ").
- These alphanumeric characters can be any of the characters whose ASCII codes fall within the range 0-255 (except the double quote character (") and carriage-return/line-feed sequences).

22

String CONSTANTS

- This range includes both the actual ASCII characters (0-127) and the extended characters (128-255).

23

Numeric CONSTANTS

- Numeric constants are positive or negative numbers. Numeric constants in BASIC cannot contain commas. The types and subtypes of constants are listed subsequently.

24

Integer CONSTANTS DEC

- One or more decimal digits (0-9), with an optional sign prefix (+ or -). The range for integer dec constants is –32 768 to +32 767.

25

Integer CONSTANTS HEX

- One or more hexadecimal digits (0-9, a-f, or A-F) with the prefix &H or &h. The range for integer hexadecimal constants is &h0 to &hFFFF.

26

Integer CONSTANTS OCT

- One or more octal digits (0-7) with the prefix &O, &o, or &. The range for integer octal constants is &o0 to &o177777.
- Note : &107 = &o107
- Also &o177777 = &hFFFF = -1

27

Long CONSTANTS DEC

- One or more decimal digits (0-9), with an optional sign prefix (+ or -) and the suffix &. The range for long decimal constants is -2,147,483,648 to +2,147,483,647.
- 212& is a long constant where as 212 is an integer constant.

28

Long CONSTANTS HEX

- One or more hexadecimal digits (0-9, a-f, or A-F) with the prefix &H or &h and the suffix &. The range for long hexadecimal constants is &h0& to &hFFFFFFFF&.

29

LONG CONSTANTS OCT

- One or more octal digits (0-7) with the prefix &O, &o,or & and the suffix &. The range for long octal constants is &o0& to &o37777777777&.

30

Fixed Point CONSTANTS

- Positive or negative real numbers (numbers containing decimal points).
- 1.2345 345.678

31

Floating Point CONSTANTS

- Positive or negative numbers represented in exponential form.

32

Floating Point CONSTANTS

- A single-precision floating-point constant is an optionally signed integer or fixed-point number (the mantissa) followed by the letter E and an optionally signed integer (the exponent).

33

Floating Point CONSTANTS

- The constant's value is the mantissa multiplied by the power of ten represented by the exponent.
- 123E+32 or 1.23E+34

34

Floating Point CONSTANTS

- Double-precision floating-point constants have the same form as single-precision floating-point constants, but use D, rather than E, to indicate the exponent.

35

SINGLE CONSTANTS

- Single-precision numeric constants are stored with 7 digits of precision (plus the exponent). Double-precision numbers are stored with 15 or 16digits of precision (plus the exponent).
- A single-precision constant is any numeric constant that has one of the following properties:
- Exponential form denoted by E
- A trailing exclamation mark (!)

36

SINGLE CONSTANTS

- A value containing a decimal point that does not have a D in the exponent or a trailing number sign (#) and that has fewer than 15 digits
- A value without a decimal point that has fewer than 15 digits but cannot be represented as a long-integer value.

37

DOUBLE CONSTANTS

- A double-precision constant is any numeric constant that has one of the following properties:
- Exponential form denoted by D
- A trailing number sign (#)
- A decimal point, no E in the exponent or trailing exclamation mark (!), and more than 15 digits.

38

SYMBOLIC CONSTANTS

- A constant may be assigned a name:

  CONST PI=3.14

39

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.

40

VARIABLES

- 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.)
- Conventions for Variable naming will be discussed later.

41

VARIABLE DECLARATION

- Simple variables can be numeric, string, or record variables. You mayspecify simple variable types in three different ways:
- Type-declaration suffix
- AS declaration statement
- DEFtype declaration statement

42

Type-declaration suffix

- Append one of the following type-declaration suffixes to the variable name:
- %, &, !, #, $.
- The dollar sign ($) is the type-declaration character for string variables.

43

Type-declaration suffix

- You can assign a string constant to the variable of up to 32,767 characters, as in the example below.
- A$ = "SALES REPORT"

44

Type-declaration suffix

- Numeric-variable names can declare integer values (denoted by a % suffix),
- long-integer values (by a & suffix)
- single-precision values (by a ! suffix),
- double-precision values (by a # suffix).

45

Type-declaration suffix

- Single precision is the default for variables without a type suffix.
- There is no type declaration character for a user-defined type.

46

Type-declaration suffix

- A% is integer type name
- A$ is string type name
- A& is long type name
- A! or A is single type name
- A# is double type name

47

AS declaration TYPE

- Declare the variable in a declaration having the 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.

48

AS declaration TYPE

- For example, the following statement declares the variable A as having a long-integer type:
- DIM A AS LONG

49

VARIABLES 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 N as integer, Nitt as Date
- Dim XYZ as STRING*30

50

VARIABLES Variable String

- 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.

51

VARIABLES Fixed String

- 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.

52

DEFtype Declaration

- Use the BASIC statements DEFINT, DEFLNG, DEFSTR, DEFSNG, and DEFDBL to declare the types for certain variable names.
- By using one of these DEFtype statements, you can specify that all variables starting with a given letter or range of letters are one of the elementary variable types, without using the trailing declaration character.

53

DEFtype Declaration

- DEFtype statements only affect variable names in the module in which they appear.
- Remember a DEFtype statement affects all variables starting with the given letter or range of letters.

54

DEFtype Example

- DEFINT A-L will make all the names starting with any letter from A-L an integer name.
- DEFSNG M-O will make all the names starting with letters M-O as single names.
- DEFSNG L will cause names starting with L as singles names.

55

USER-DEFINED EXAMPLES

```
TYPE Result
    Name as String*20
    Phy as Byte
   Chem as Byte
    Math as byte
   Total as Integer
    PorF as String*4
END TYPE

DIM  FSC as Result
```

56

USER-DEFINED EXAMPLES

```
TYPE Triangle
    Side1 as single
    Side2 as single
    Side3 as single
    Area as double
    Perimeter as Single
END TYPE

DIM  RightT  as TRaingle
```

57

USER-DEFINED EXAMPLES

RightT.Side1
RightT.Side2
RightT.Side3
can be assigned single values

Where as RightT. Area can only be given a double value.

58

VARIABLES

- Simple variables refer to a single number, string, or record. (SCALAR)

  DIM X As Long
- Array variables refer to a group of objects, all of the same type. (VECTOR)

  DIM X(20) as Long, Y(10,20) as Double

59

VARIABLES

- A numeric variable, whether simple or array, can be assigned only a numeric value (either integer, long integer, single precision, or double precision).
- A string variable can be assigned only a character-string value.

60

VARIABLES

- A constant value: A = 4.5
- The value of another string or numeric variable:

1. B$ = "ship of fools"
2. A$ = B$
3. Profits = NetEarnings

61

VARIABLES

- You can assign one record variable to another only if both variables are the same user-defined type.
- However, you can always assign individual elements of a record to a variable of the corresponding type.

62

Suppose the following is an extract from a program

- TYPE EmployeeRec
  FullName AS STRING * 25
  OverTime AS Single
  END TYPE
- DIM CurrentEmp AS EmployeeRec
  DIM XYZ as single
  DIM LastEmp AS EmployeeRec

63

You can do the following:

- XYZ=CurrenEmp.OverTime
- LastEmp=CurrentEmp

64

Variable NAMES

Convention

- 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.

65

Variable NAMES

Convention

- 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)
- However, the following statement is legal:
- TimeLog = 8

66

Variable NAMES

Convention

- Reserved words include all BASIC commands, statements, function names, and operator names.
- Note: Variable names cannot duplicate procedure (SUB and FUNCTION) names or symbolic constant (CONST) names.

67

Summary

Suffix	AS Type	DEF Type	Variable Type	SIZE
%	Integer	DEFINT	Inetger	2
&	Long	DEFLNG	Long	4
!	Single	DEFSNG	Single	4
#	Double	DEFDBL	Double	8
$	String	DEFSTR	String	2+LEN

Fixed length string takes 1 B /Character

68