Visual Basic 6.0 Language Summary

Please note that this list is not complete; various categories of commands not covered in class have been omitted here (for example, financial functions, commands used to store and retrieve random and binary files, etc.). An additional short list of graphics functions will be provided later.

Special Characters

#	Double operator; sets variable to Double type.
#	Used to enclose a Date data type (eg #2/22/99#)
$	Sets type of a variable to string (eg a$="Hello!")
%	Sets type of a variable to integer (eg a%=5)
&	String concatenation operator (eg c$=a$&b$)
'	Remark - rest of line is ignored by compiler (like Rem)
*	Multiplication operator (eg c=a*b)
+	Addition operator (eg c=a+b)
-	Subtraction or negation operator (eg x=-1; c=b-a)
/	Division operator (eg c=a/b; c=1.2/2.34)
<	Less-than operator (e.g. if a<b then...)
<=	Less-than or Equal-to operator
<>	Non-equal operator (eg Do Until a<>b...)
=	Equality operator (eg c=a*b)
<	Greater-than operator (eg If c>b Then...)
>=	Greater-than or Equal-to operator
?	Print (eg ?a prints current value of a in Immediate window)
\	Integer division operator (like / but returns integer)
^	Exponent operator (eg c=a^b means a to the b power)

VB Keywords, Statements and Commands

Abs	Absolute value (eg b=Abs(a))
And	Logical AND operator
Asc	Returns ASCII code of (first letter of) a string (eg z=Asc("a")
Atn	Returns arctangent of a number in radians (eg z=atn(x))
Beep	Produces beep sound
Call	Executes a user-defined procedure (eg Call Testit(z)) where z is an optional parameter; can provide multiple parameters be passed
CByte	Converts to Byte data type
CDbl	Converts to double precision floating point number
CDec	Converts to variable type Decimal
ChDir	Change default directory to specified pathname (eg ChDir(C:\")
Chr	Returns the character string of the ASCII value argument (eg x$=Chr$(66))
CInt	Converts the argument to an integer (eg CInt(x) returns 1 if x=1.25)
CLng	Converts the argument to a Long integer (eg CLng(200000.5) returns 200000)
Close	Closes the file specified by the argument (a file number between 1 and 255)
Const	Declares a value as a constant
Cos	Returns the cosine of an angle specified in radians
CSng	Converts the argument to a single precision floating point number eg x=CSng(y#)
CStr	Converts the argument to a string (eg x$=CStr(1.234)
CVar	Converts the argument to a Variant data type
Date$	Returns current date as a string
DefDbl	Specifies default data type for arguments to Double (eg DefDbl X would make all new untyped variables beginning with X a Double type.

DefDec	Also, DefInt, DefLng, DefSng, DefStr, DefVar all specify default data types for arguments as integer, long integer, string, or variant, as for DefDbl above
Dim	Defines a variable (with As type identified)
Do	(with Loop) Cycles through a loop until the specified condition is met; Do...While, Do...Until (to test at top of loop) or Loop...While and Loop...Until (to test at bottom of loop)
End	Stops execution or ends definition of a function or subroutine
EOF	Returns the end-of-file condition for the specified file number (eg Do Until EOF(1))
Exit	Terminates an operation before conditions are complete, as in a loop (eg If x=1 Then Exit Sub)
Exp	Raise the base of the natural logarithm e to the (argument) power (eg y=exp(x))
False	Logical boolean value of false
FileCopy	Copy a file from the source to the destination (format is FileCopy source$, destination$, where these strings specify paths and filenames)
FileLen	Returns the length, in bytes, of the specified file (format is FileLen(filename$))
Fix	Returns the integer of a specified value (works like Int but behaves slightly differently for negative numbers)
For	(For...To...Next) Loops until the end value parameter (eg To endval) is reached (syntax, For cntval = startval to endval Step incval...(statements)...Next cntval)
Format	Generates output in a desired format with specified digits to right of decimal, leading zeros, special characters like $, dates, etc. (syntax is Format(numeric expression,editPattern), eg Format(x,"##.###")
Function	(Function... End Function) Creates a function in a module or a form; the function definition can indicate types of arguments to be received, and types to be returned. Syntax, Function function-name(arguments)
Global	Declares a variable available to all forms and modules in that project (can be used only in a code module; use Public to make method or property of a form available to the whole project.
GoSub	(GoSub...Return) Jumps to a piece of code outside function or subroutine, then returns when done to next line of code. Syntax, GoSub LineLabel; elsewhere, LineLabel: (statements)... Return
GoTo	Jumps to a piece of code outside function or subroutine, but does not return (syntax, GoTo lineLabel)
Hex$	Function generates a string containing hexadecimal representation of the numeric argument (eg Print Hex$(x))
If...Then	(If...Then...ElseIf...End If) Conditional execution of statements based on evaluation of conditions that can change during program execution. Syntax: If condition1 Then... [actions1] ElseIf conditions2 Then... [actions2] Else... [more actions] End If
Input #	Returns a string from an Input file, directly into a specified variable. Syntax: Input #filenumber, variable1 (,variable2, etc optional)
InputBox$	Produces an input dialog box allowing entry of text information (similar to a message box) Syntax: InputBox$(msg$[,[title$][,default$][,xpos%,ypos%]]]) where default$ is an optional string to show in the input box. Returns a string
InStr	Returns the first location within a string where a substring appears. Syntax: InStr([startpos&,]string$,substring$)
InStrRev	Returns the first location within a string where a substring appears from the end of that string. Syntax: InStrRev(string$,substring$,[startpos&])
Int	Converts a number to the Integer data type (eg x%=Int(y))
Kill	Delete a specified file. Syntax: Kill filename$
LBound	Returns the lowest subscript available in an array. Syntax: LBound(arrayname[,dimension%])
LCase$	Returns an all-lowercase string. Syntax: Print LCase$(string1$)

Left$	Returns a string containing a specified amount of the left portion of a string. Syntax: Left$(string1$,Length%)
Len	Returns the length of a specified string. Syntax x%=Len(String1$)
Let	Sets a variable equal to a particular value. Implied by =. eg Let x=1
Line Input #	Reads a single line from a (sequential) file. Line is delimited by a carriage return. Syntax Line Input #filenumber%, variable$
Loc	Sets the current read or write position with the open file. Can be used to track the expanding size of a file stored to a disk. (The position value changes with every read or write to a file.)
LOF	Returns length of a currently open file. Syntax: x=LOF(filenumber%)
Log	Returns the natural log of the argument (eg y=log(x))
Me	Return to currently active form.
Mid$	Returns a substring of the string argument, of a specified length and start position. (eg b$=Mid$(a$,start%[,Length%]). Also allows replacement of a substring within a string.
MkDir	Creates a new directory. (eg MkDir "C:\VBSamples\VBdemo1")
Mod	The modulo arithmetic operator, gives the remainder of a division (eg if B=7 and C=2, B Mod C yields 1.
MsgBox	Displays a dialog box containing information, and may return information about user selection. Both a function and a statement.Selections for OK, cancel, yes, no, retry, abort, ignore. Syntax: MsgBox(message$[,boxtype%,windowtitle$]) (eg Call MsgBox("Danger!" and a=MsgBox("Choice?",2,"Proceed?")
Name	Renames a file, directory or folder. Syntax: Name oldname As newname
Now	Returns current date and time (related functions include Day, Hour, Minute, Month, Second, Weekday, Year, Time, TimeValue, Date
Oct$	Converts argument to a string representation of its octal (number base 8) equivalent. (eg Print Oct(64))
Open	Opens a disk file for reading or writing. Syntax: Open filename$ [for mode] [Access access] [locktype] As #filenumber where mode = Input, Output or Append (also Binary, Random), access = Read, Write or Read Write, and locktype= Shared, Lock Read, Lock Write and Lock Read Write
Option Base	Sets default lower bounds for arrays (eg Option Base 1)
Or	Logical Or (eg If a=1 Or a=7 then...)
Print #	Writes string data to a specified file. Syntax: Print #filenumber,[Spc(n) or Tab(m)] expression [,]
Private	Makes a variable's scope private to a particular form, module or routine (eg Private [function or subroutine or variable]
Public	Makes a variable's scope accessible from outside a form or module
QBColor	Returns a standard (Long) color value from a QuickBasic color. These range from black(0) to white (15), with blue=1, green=2, red=4
Randomize	Initializes the seed of the random number generator; use before Rnd() function
Rem	Remark, makes all text on rest of line ignored by compiler (also ')
Return	Returns from a GoSub routine to next line of code following GoSub
RGB	Returns a long value for color representing the three RGB values passed to it. Syntax: RGB(red%,green%,blue%) (eg RGB(0,255,0))
Right$	Returns a string containing a right portion of the string. Syntax: Right$(string$,Length&)
Rnd	Function returns a Single precision number between 0 and 1
Round	Rounds off the number in the argument to the specified number of decimal places. Syntax: Round(num,decplaces) (eg Round(a,2) where a=1.25587 will yield 1.26)
SavePicture	Saves the graphic stored in a Picture or Image control. Syntax: SavePicture objectReference, picturefile$ where objectReference is to the Picture property of a control. (eg SavePicture Picture1.Picture,"c:\myPict.bmp")

Select Case	(Select Case testexpression...Case expression1... statements... Case expression 2.. statements... Case Else... statements... End Select) Executes a group of statements when an expression equals a test expression. Like If...Then...Else, will compare expression against multiple values. Case Else statement is executed if none of the values matches.
Sgn	Returns the sign, + or -, of the number argument (+1, -1, or 0 if argument is 0)
Sin	Returns the sine of an angle, expressed in radians
Space$	Returns a string with number of specified spaces
Spc	Adds spaces for formatting, specifically within Print and Print # commands
Split	Splits a string into a number of smaller strings between specified delimiters. Syntax: Split(string$ [,delimit[,count]]) where delimit is the delimiting string (assumes a space if omitted) and count is number of strings to be returned. (eg if a ="This is a test" then Split(a) will give a(0)="This" and a(4)="Test")
Sqr	Returns the square root of the number argument. (eg x=squ(4) would yield x=2)
Static	Makes a variable that is local to a particular routine retain its value; the next execution of the routine can thus access the last value. Syntax: Static name As type [,name [As type]]...
Stop	Halts execution of a program. A semipermanent breakpoint that would be saved with a program.
Str$	Converts a numeric expression to a string (eg a$=str$(1.234))
StrReverse	Reverses the string argument character by character.
String$	Creates a string of a repeating specified character. Syntax: String$(num-of-characters, character$)
Sub	(Sub... End Sub) Definition of a subroutine, optionally including arguments to be received when called. Syntax: Sub subname (arguments)... [statements]... End Sub
Tab	Adds a tab to formatting in a Print or Print # statement. Syntax: Tab(column%)
Tan	Returns the tangent of an angle expressed in radians.
Time$	Retrieves the current time from the system.
Timer	Returns the number of seconds elapsed since midinght.
UBound	Returns the highest subscript available in an array. Syntax: UBound(arrayname[,dimension%])
UCase$	Returns a string completely converted to upper case characters
Unload	Unloads a form or control object from memory
Val	Returns a numeric value contained in a string; if there is no numeric value in the string, a 0 is returned.
Value Property	Holds the value for a specific control. For example, in a scroll bar control, the Value property = the current thumb position. Syntax: object.Value
While... Wend	Cycles through a loop of code until the necessary condition is met. Syntax: While condition... [statements]... Wend
Width #	In formatting to a file or a printer, sets width between 1 and 255. Syntax: Width #filenumber, width%
Write #	Writes data to a specified file. Syntax: Write #fileNumber , var1, var2, etc.
Xor	Exclusive Or, logical; bit by bit, if both bits are 0 or 1, the result is zero; if either bit is 0 and the other 1, the result for that bit is 1.