Chapter 9 - File Handling (pgs. 357 - 381)

There are three basic types of files used in Visual Basic:

* Random - Data organized into records(a complete set of data for a single entry)
* Sequential - Used as if it is printed output
* Binary - Information is arranged to give byte-by-byte access to the user

Open Statements - Tells VB whether you will use the file as a random, sequential, or binary file

**Refer to the textbook to learn how to construct and use an open statement**

Put statement - Writes data to a file opened as a binary or random file

* Random files - following the file number is the record number of the data being put into the file
* Binary files - the number following the file number is the byte position of the data being put into the file

**Refer to the textbook to learn how to construct and use a put statement**

Get statement - Reads data from a file opened as either a random or binary file, matches the put statement

* Random files - *record number* refers to the position in the file where the record whose number is *record number* is stored
* Binary files - *record number* refers to a byte offset from the beginning of the file, the data at this location is read from the file and stored in *variable*

**Refer to the textbook to learn how to construct and use a get statement**

Close statement - Without any parameters, the close statement closes all open files

Kill Statement - Deletes a file whose pathname is specified

EOF Function - Stands for End Of File, returns the value of True when the Get command has failed to get information from the file because the file is empty

With Statement - Simplifies access to the field of records

**Refer to the textbook to learn how to construct and use these statements**

Record Arrays

* A record is composed of one or more logically related variables
* To bind these variables, create a user-defined data type - combines one or more variables of various types into a single data type
* You must define it before you use it
  + “Type” then name of the data type
  + Each item in the body of the type definition is a field
  + These definitions appear in the general declarations section of a code module
  + Any variables you declare here are global

**Refer to the textbook to learn how to dim and assign values for record arrays**

**Refer to the textbook for the Phonelist program**

Error Handling

* Run-time errors occur when something interrupts the execution of a program
* Error trapping - redirecting the flow of execution when errors occur
* On Error Resume Next - traps an error by executing the statement following the one that caused the error
  + The action that caused the error is never completed
* On Error GoTo line - redirects the flow of execution to an error-handling routine
  + Error handling routine - A piece of code that either resolves the error and lets the program resume its normal course or displays information about the error so that it may be corrected
* Flow is redirected by specifying a line number or a line label
  + Line label is an identifier up to 40 characters long, followed by a colon
    - Must be first entry in a line
* Exit Sub statement interrupts the flow of program execution and ends the procedure being executed
* Error-Handler - A piece of code within the procedure, usually at the end of the procedure
  + Not executed unless there are errors

**Refer to the textbook to learn how to construct and use these statements**