1. File Types

In addition to getting data from the keyboard and sending data to the screen, a program also can get data from and send data to a file on a disk. Getting data from a file is referred to as “reading the file,” and sending data to a file is referred to as “writing to the file”. Files to which data is written are called **output files**, and files that are read by the computer are called **input files**. Most input and output files are composed of lines of text that are written and read sequentially (in consecutive order, one line at a time). Such files are referred to as sequential access files (also text files, since they store text). You can also create random access and binary access files, which let you access data in random order and according to their byte locations, respectively

**Create File Objects**

The iostream file contains the definitions of the istream and ostream classes from which the cin and cout objects, respectively, are created. You do not have to create the cin and cout objects in a program because C++ creates the objects in the iostream file for you. Objects are also used to perform file input and output operations in C++, but they must be created by the programmer. To create a file object in a program, the program must contain the #include <fstream> directive. The fstream file contains the definitions of the ifstream (input file stream) and ofstream (output file stream) objects, which allow you to create input file objects and output file objects. Although not required, it is useful to begin input file object names with “in” and output file object names with “out”, so as to distinguish a program’s input file objects from its output file objects.

**The Process of File Input/Output**

File I/O is a five-step process

**Include header file fstream in the program.**

#include <fstream>

**Declare file stream variables**

ifstream inData;

*declares* inData *to be an input file stream variable*

ofstream outData;

*declares* outData *to be output file stream variable*

**Associate the file stream variables with the input/output sources**.

*This is also called opening the files. The stream member function open is used to open files*.

*The syntax for opening a file is*:

fileStreamVariable.open(sourceName[, mode])

*Suppose you include the declaration from step 2, the following statements associate* inData *with* prog.txt *and* outData *with* prog.out*.*

inData.open (“prog.txt”);// **open the input file**

outData.open(“prog.out”);// **open the output file**

**Use the file stream variables with >>, <<, or other/output functions**.

*The syntax for using the >> or << with file stream variable is exactly the same as the syntax for using* cin *and* cout*. Instead of using* cin *and* cout*, however, you use the file stream variable names that were declared. For example, the statement:*

inData >> payRate;

*reads the data from the file* prog.txt *and stores it in the variable* payRate*. The statement:* outData << “The paycheck is: ” <<pay<< endl;

stores the output – The paycheck is: 565.78 – in the file prog.out. This statement assumes that the pay was calculated as 565.78.

**Close the files**

inData.close();

outData.close();

Closing the file means that the file stream variables are disassociated from storage area and are freed. Once they are freed, they can be reused for other file I/O.

|  |  |
| --- | --- |
|  | **Example** |

|  |
| --- |
| A program needs to read a sequential access files, line by line, and display each line on the computer screen. The file, which was opened successfully, is associated with the infile object. Write the code to read and then close to the file.  string textLine = "";  getline(inFile, textLine);  while (!inFile.eof())  {  cout << textLine << endl;  getline(inFile, textLine);  } //end while  inFile.close(); |

|  |  |
| --- | --- |
|  | **Activity** |

1. Sequential Access Files

|  |
| --- |
| You are given a file consisting of students’ names in the following form: lastName, firstName middleName. (Note that some students may not have middle names). Write a program that converts each name to the following form: firstName middleName lastName. Your program must read each student’s entire name in a variable and must consists of a function that takes as input a string, consists of student’s name, and returns the string consisting of the altered name. Use the string function find to find the index of , ; the function length to find the length of the string; and function substr to extract the firstName, middleName, and lastName |

1. Test your program with a file consisting of the following names:

Miller, Jason Brian

Blair, Lisa Maria

Gupta, Anil Kumar

Arora, Sumit Sahil

Saleh, Rhonda Beth