**LaGuardia Community College – Last Update**

08

**Fall**

Part 11: File Processing

Dr. Andi Toce

Lecture Notes for MAC 101 (Introduction to Computer Science)

Last updated / viewed: May 12, 15

Table of Contents

[1. Output Stream Objects 2](#_Toc419207875)

[2. Input Stream Objects 4](#_Toc419207876)

[3. Append at the End of a File 5](#_Toc419207877)

# Output Stream Objects

The output generated by the C++ program will be lost unless is stored somewhere in permanent memory. In this section we will discuss how to store and retrieve data from a file.

First Example: Creating a new file in the current workspace.

|  |
| --- |
| FileProcessingExample1.cpp |
| #include <fstream>  using namespace std;  int main() {  ofstream myOutputFile("output.txt");  myOutputFile << "Andi Toce" << endl;  myOutputFile << "MAC 101" << endl;  myOutputFile.close();  return 0;  } |

**Note**: File stream objects support the same function calls as the cin/cout objects.

**Try Now:** Change the output text and run the program again. What do you notice?

**Try Now:** Can we open multiple file streams simultaneously? Add a new ofstream object and print something in it. Does it work? Can two stream objects write in the same file?

Second Example: Creating a new file in a new, specific location.

|  |
| --- |
| FileProcessingExample2.cpp |
| #include <fstream>  using namespace std;  int main() {  ofstream myOutputFile("c:\\TestFolder\\output2.txt");  myOutputFile << "Andi Toce" << endl;  myOutputFile << "MAC 101" << endl;  myOutputFile.close();  return 0;  } |

**Question:** Why do we need to use double slash (\\) in the file path?

Third Example: Create/ open user specified file name.

|  |
| --- |
| FileProcessingExample3.cpp |
| #include <iostream>  #include <fstream>  #include <limits.h>  using namespace std;  int main() {  char filename[PATH\_MAX+1];  cout << "Enter a file name and press ENTER: ";  cin.getline(filename, PATH\_MAX);  ofstream file\_out(filename);  if (! file\_out) {  cout << filename << " could not be opened."<< endl;  return -1;  }  cout << filename << " was opened." << endl;  file\_out << "Andi" << endl;  file\_out << "Toce" << endl;  file\_out.close();  return 0;  } |

**Question**: What is the length of MAX\_PATH and why do we use MAX\_PATH +1 for the array size?

**Try now:** Rewrite FileProcessingExample3.cpp so it prompts for directory location and filename separately. (Hint: Use two strings and use the **strcat** function to join them.)

**Try now:** Write a program that lets the user enter any number of lines of text, one at a time. In effect, this creates a primitive editor, which permits text entry but no editing of a line of text after it’s been entered. Set up a loop that doesn’t terminate until the user presses ENTER without typing any text (a zero-length string).

Alternatively, you can recognize a special code (for example, @@@) to terminate the session. You can then use the **strcmp** (“string compare”) function to detect this string.

if (strcmp(input\_line, "@@@"))

break;

Remember to print a short prompt before each line of text, such as the following:

Enter (@@@ to exit)>>

# Input Stream Objects

|  |  |
| --- | --- |
| ReadFile.cpp | Output |
| #include <iostream>  #include <fstream>  using namespace std;  #define COL\_WIDTH 80  #define MAX\_PATH 200  int main() {  int c; // input character  int i; // loop counter  char filename[MAX\_PATH + 1];  char input\_line[COL\_WIDTH + 1];  cout << "Enter a file name and press ENTER: ";  cin.getline(filename, MAX\_PATH);  ifstream file\_in(filename);  if (! file\_in) {  cout << filename << " could not be opened.";  cout << endl;  return -1;  }  while (true) {  for (i = 1; i <= 24 && ! file\_in.eof(); i++) {  file\_in.getline(input\_line, COL\_WIDTH);  cout << input\_line << endl;  }  if (file\_in.eof())  break;  cout << "More? (Press 'Q' and ENTER to quit)";  cin.getline(input\_line, COL\_WIDTH);  c = input\_line[0];  if (c == 'Q' || c == 'q')  break;  }  return 0;  } | Enter a file name and press ENTER: andi.txt  Andi  Toce |

**Try now:**

1. Create a folder PracticeFolder in the c:\ directory of your computer.
2. Modify the appropriate C++ program from above to create a new file “numbers.txt”.
3. Write in the file “numbers.txt” the first 100 positive integers. Each integer using a new line.
4. Use the program ReadFile.cpp to output the content of the “numbers.txt” file 10 numbers at a time.

# Append at the End of a File

|  |
| --- |
| WriteFileAppendExample.cpp |
| #include <iostream>  #include <fstream>  #include <string>  #include <cstdlib>  using namespace std;  int main() {  int id=0;  string name;  char grade;  ofstream classroomRecords("records.txt", ios::*app*);  if (! classroomRecords) {  cerr << "File could not be opened." << endl;  exit(1);  }  cout << "Enter ID, student name and grade: " << endl;  cout << "Enter end-of-file (^z) to end input: \n --> " << endl;  while (cin >> id >> name >> grade){  classroomRecords << id << " " << name << " " << grade << endl;  cout << "--> ";  }  return 0;  } |

**Homework:** Write a C++ program that allows the user to search a file “MyText.txt” located in the current workspace for any word (String). The program should print the result of the search. In the event the word is found the program should indicate the line where it was found in the text file.

Example:

Enter the word you are looking for: **Andi**

The word Andi was not found!

Or

Enter the word you are looking for: **Andi**

The word Andi was found in line 3 of the text file!