John Rollinson

CS151

Spring 2020

Module 9; Fill-in-the-blank and Review Questions:

5. The three file stream data types are **fstream**, **istream**, and **ostream**.

15. If a file fails to open, the file stream object will be set to **NULL**.

39. Design a pseudocode algorithm for determining the length of a file (the number of bytes that are stored in the file).

* Open file stream for reading the number of bytes from the file using ifstream class or fstream class objects.
* Set the read position to the end of the file using the seekg function of the file stream class **file.seekg(0L, ios::end)**

Where the file is an object of the file stream, the seekg is a member function, 0L is the position in a file and ios::end sets the read position to the end of the file.

* Now using tellg which is a member function of the file stream class gets the position from where to get the position which returns the current position in the file. This is called by using the file stream object.

**Long int bytes = file.tellg();**

Where bytes is a variable which is used to store the value returned by tellg() function.

* Print the value to the console:

cout << “No. of bytes in the file: “ << bytes;

* End program.

40. Design a pseudocode algorithm for comparing two files to see if their contents are identical.

* Open two files by using two file stream objects in read mode.
* Calculate the size of the two files by using tellg member function of file stream. Check if two files sizes are equal then proceed.
* Loop and get a character from the first file.
* Get a character from the second file.
* Check if the two characters are equal or not.
* If they are not equal, if they are not then print to the file.
* Increment the count value.
* End of loop iteration.
* Check the counter value with the size of the file and if they are equal then print their content.
* End of program.

41. Design a pseudocode algorithm for reversing the content of one text file into another file. Assume that the amount of memory is limited, so that you cannot read the entire source file into memory before you start writing it into the second file in reverse order.

* Open a file in input mode
* Open another file in output mode
* Set the position of the file to the end of the file using the fseekg function of the file stream.
* Read the last character from the first file using the get function from the file stream object.
* Write the character to the output file.
* Check if the memory is available or not for writing using the fail() function.
* If the fail() function returned true then display and error message for writing to the file.
* Else write the character to the output file.
* End of program.