# Chapter 10

## True or False

1. When working with a sequential access file, you can jump directly to any piece of data in the file without reading the data that comes before it.

False

2. In most languages, when you open an output file and that file already exists on the disk, the contents of the existing file will be erased.

True

3. The process of opening a file is only necessary with input files. Output files are automatically opened when data is written to them.

False

4. The purpose of an EOF marker is to indicate where a field ends. Files typically contain several EOF markers.

False

5. When an input file is opened, its read position is initially set to the first item in the file.

True

6. When a file that already exists is opened in append mode, the file’s existing contents are erased.

False

7. In control break logic, the program performs some ongoing task (such as processing the items in a file), but permanently stops the task when a control variable reaches a specific value or changes its value.

False

## Short Answer

1. Describe the three steps that must be taken when a file is used by a program.

Open file  
Process file  
Close file

2. Why should a program close a file when it’s finished using it?

It disconnects the program from the file and prevents data loss.

3. What is a file’s read position? Where is the read position when a file is first opened for reading?

Read position marks the next item that will be read from a file. When opened for reading this is set to the first item in the file.

4. If an existing file is opened in append mode, what happens to the file’s existing contents?

Existing content remains unchanged, unless you change it after opening the file.

5. In most languages, if a file does not exist and a program attempts to open it in append mode, what happens?

The file is created

6. What is the purpose of the eof function that was discussed in this chapter?

EOF shows where the file contents end. It’s used in this chapter to loop through a file of unknown length.

7. What is control break logic?

A way to interrupt processing when the value of a specific variable changes, allowing custom action to be taken.

## Algorithm Workbench #3

Design an algorithm that does the following: opens an output file with the external name number\_list.dat, uses a loop to write the numbers 1 through 100 to the file, and then closes the file.

Open file  
Declare int n  
For n = 1 - 100  
 Write n to file  
End For   
Close file

OR

std::ofstream numFile("path/to/numbers.txt");  
for ( int n = 1; n < 101; n++ ) {  
 numFile << n << " ";  
}  
numFile.close();  
  
std::ifstream numbers("numbers.txt");  
char nbr;  
while( ! numbers.eof() ) {  
 numbers.get(nbr);  
 std::cout << nbr;  
}  
numbers.close();