

---

**CPT 111 – PRINCIPLES OF PROGRAMMING**  
**WEEK 6: PROGRAMMING LAB**  
**File Processing**

Answer the following questions: 1, to 6, 9, 10.

1. Write a program that asks the user for the name of a file. If no such file exists, the program should display an error message; otherwise it should display the alternate lines in the file (the lines with the numbers 1, 3, 5 ...). The corresponding line number should be displayed before each line.

Note: create a simple text file to test this program.

2. Write a program that asks the user for the name of a file. The program should display the contents of the file on the screen. If the file's contents won't fit on a single screen, the program should display 24 lines of output at a time, then pause. Each time the program pauses, it should wait for the user to strike a key before the next 24 lines are displayed.

Note: create a simple text file to test this program.

3. Write a program that reads and prints a joke and its punch line from two different files. The first file contains a joke, but not its punch line. The second file has the punch line as its last line, preceded by "garbage". Your program should open the two files then do the processing. The program should read and display each line in the file it is passed (the joke file). The second function should display only the last line of the file it is passed (the punch line file). It should find this line by seeking to the end of the file and then backing up to the beginning of the last line.

Data to test your program can be found in joke.txt and punchline.txt files.

4. Write a program that asks the user for the name of a file. The program should display the following data about the file:
  - Number of words (each word is preceded by a blank space or newline.)
  - Number of sentences (Each sentence is terminated by a period.)

Create your own file to test the program

5. Write a program that asks the user for the name of a file. The program should display the contents of the file on the screen. Each line of screen output should be preceded with a line number, followed by a colon. The line numbering should start at 1. Here is an example:

```
1:George Rolland  
2:127 Academy Street
```

3:Brasstown, NC 28706

If the file's contents won't fit on a single screen, the program should display 24 lines of output at a time, then pause. Each time the program pauses, it should wait for the user to strike a key before the next 24 lines are displayed. (You can modify the program from Q2)

6. Write a program that asks the user for the name of a file. If the file cannot be opened, the program should display an error message; otherwise the program should display only those words in the file that have at least 10 characters in them. The program should also display the number of characters alongside each word. As an end note, the program should display a total number of words printed. The program should close the file before exiting.
7. File encryption is the science of writing the contents of a file in a secret code. Your encryption program should work like a filter, reading the contents of one file, modifying the data into a code, then writing the coded contents out to a second file. The second file will be a version of the first file, but written in a secret code. Although there are complex encryption techniques, you should come up with a simple one of your own. For example, you could read the first file one character at a time, and add 10 to the ASCII code of each character before it is written to the second file.
8. Write a program that decrypts the file produced by your program in Q7. The decryption program should read the contents of the coded file, restore the data to its original state, and write it to another file.
9. Consider the following incomplete C++ program:

```
#include <iostream>
int main()
{
    ...
}
```

  - a. Write a statement that includes the header files `fstream`, `string`, and `iomanip` in this program.
  - b. Write statements that declare `inFile` to be an `ifstream` variable and `outFile` to be an `ofstream` variable.
  - c. The program will read data from the file `inData.txt` and write output to the file `outData.txt`. Write statements to open both of these files, associate `inFile` with `inData.txt`, and associate `outFile` with `outData.txt`.
  - d. Suppose that the file `inData.txt` contains the following data:  
Giselle Robinson Accounting  
5600 5 30  
450 9  
75 1.5  
The first line contains a person's first name, last name, and the department the person works in. In the second line, the first number represents the monthly gross

salary, the bonus (as a percent), and the taxes (as a percent). The third line contains the distance traveled and the traveling time. The fourth line contains the number of coffee cups sold and the cost of each coffee cup. Write statements so that after the program executes, the contents of the file outData.txt are as shown below. If necessary, declare additional variables. Your statements should be general enough so that if the content of the input file changes and the program is run again (without editing and recompiling), it outputs the appropriate results.

```
Name: Giselle Robinson, Department: Accounting
Monthly Gross Salary: $5600.00, Monthly Bonus: 5.00%,
Taxes: 30.00%
Paycheck: $4116.00
Distance Traveled: 450.00 miles, Traveling Time: 9.00
hours
Average Speed: 50.00 miles per hour
Number of Coffee Cups Sold: 75, Cost: $1.50 per cup
Sales Amount = $112.50
```

- e. Write statements that close the input and output files.
- f. Write a C++ program that tests the statements in parts a through e.

10. The manager of a football stadium wants you to write a program that calculates the total ticket sales after each game. There are four types of tickets—box, sideline, premium, and general admission. After each game, data is stored in a file in the following form:

```
ticketPrice      numberOfTicketsSold
...
```

Sample data are shown below:

```
250  5750
100  28000
50   35750
25   18750
```

The first line indicates that the ticket price is \$250 and that 5750 tickets were sold at that price. Output the number of tickets sold and the total sale amount. Format your output with two decimal places.

11. Three employees in a company are up for a special pay increase. You are given a file, say Ch3\_Ex6Data.txt, with the following data:

```
Miller Andrew 65789.87 5
Green Sheila 75892.56 6
Sethi Amit 74900.50 6.1
```

Each input line consists of an employee's last name, first name, current salary, and percent pay increase. For example, in the first input line, the last name of the employee is Miller, the firstname is Andrew, the current salary is 65789.87, and the pay increase is 5%. Write a program that reads data from the specified file and stores the output in the file Ch3\_Ex6Output.dat. For each employee, the data must be output in the following form: firstName lastName updatedSalary. Format the output of decimal numbers to two decimal places.