Computer Science 1	Exercises 17.01-08	Date:
Name:		Period:

1. What is the most fundamental justification for using a computer?

2. How many *bits* are in a *byte*?

3. What is a *field*?

4. What is a *record*?

5. A *Data Base File* consists of a _____.

6. From where can *input streams* come?

7. To where can *output streams* go?

8. What is a *file data structure*?

9. What is an *external file*?

10. What is the difference between a *text file* and a *binary file*?

11. What is the difference between a *sequential access file* and a *random-access file*?

12. Look at **TextFiles01.py** and refer to the "File Data Structure Definition".
What are the "internal data structure" and the "external file name" in this program?

13. The **open** function has 2 arguments. The first is the name of the *external* file.
What does it mean if the second argument is the letter **'W'**?

14. Compare and contrast Python's **print** and **write** commands.

15. When you are done with a file, what should you do?

16. Compare and contrast Python's **input** and **readline** commands.

17. What Python function lets you know if a certain file exists?

18. Look at program **TextFiles04.py** and its file output. Even though there were 5 separate **write** commands, all of the output is together on the same line. Why did this happen?

19. Look at program **TextFiles05.py** and its file output. How did this program cure the problem of the previous program?

20. Look at program **TextFiles06.py** and its output. There are 5 lines of text in the file **"TextFiles05.txt"**. Why does the program only display the first one?

21. Look at programs **TextFiles05.py** and **TextFiles06.py** again. Why does the chapter say that the latter program will crash if the former program has not been executed?

22. Look at program **TextFiles07.py** and its output. This program fixes the issue of the previous program with a **for** loop, but in the process, it creates 2 other issues. What are they?

23. Look at **TextFiles08.py**. Explain how this program fixes one of the issues of the previous program.

24. In many languages, reading past the end of the file causes a run-time error. What happens in Python?

25. Look at **TextFiles10.py**. Explain how this program fixes the other issue from program **TextFiles07.py**.

26. In Python, the _____ command will read the entire file and store everything in an array of strings.

27. Look at program **TextFiles12.py** and all of its outputs. Explain how it is possible that this program actually displayed itself.

28. Refer to the previous question. While the program did display itself, the indenting is gone. Why is this?

29. If you want to write numbers to a text file, what must you do first?

30. Look at programs **TextFiles14.py**. What is wrong with this program?

31. Look at programs **TextFiles15.py**. How did this program cure the problem of the previous program?

32. Look at programs **TextFiles16.py**. What is wrong with this program?

33. Look at programs **TextFiles17.py**. How did this program cure the problem of the previous program?

34. Can a text file contain different types of data (like strings and numbers in the same file)?

35. Can a program work with more than one file data structure at the same time?

36. In terms of files, what does *appending* mean?

37. Are text files required to have a **.txt** extension?

38. How are text files useful in graphics programs?