**Csci 1523**
**1523 Study Guide Module 06 - Text Files and Exceptions**

**Student (Print):** Amethyst O'Connell

---

This study guide contains 6 pages (including this cover page) and 35 problems. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your initials on the top of every page, in case the pages become separated.

You may use your books, notes, calculator or internet sources while completing this study guide.

Please try to answer the sections clearly and PRINT your answers legibly.

*Special Note:* The completion of study guides in this course is to supplement your learning of the materials they are not required as a part of normal grading. However they may enhance the extra credit portion of the course if attendance has been regular and laboratories completed satisfactorily.

**Chapter 8 Dierbach Study Guide**

To this point in the course we have used volatile and non-persistent storage (main memory) only. During this module we will begin to work with files. In files data persists between executions of the program and is made available to other programs which may have need to use it.

In Python we use two basic types of files:

1. **Text files:** which are formatted in an ASCII compatible format and are processed by reading the data contained in them in a sequential manner.
2. **Direct access files:** which are formatted in binary form and are processed by accessing records contained in them directly.

For our purposes we will only work with ASCII text files. These files are convenient and can be readily edited with most common text editors found on various operating systems.

Below find a set of questions concerning the use of text files in the Python language. Use your text or other resources when answering each of these.

1. Data stored in memory while a program is executing is stored in Random Access memory.
2. Data stored to secondary storage such as a hard disk drive while a program is executing is stored in Hard Disk memory.
3. A text file is a file containing characters structured as lines of text.
4. The individual lines of code contained in a text file are terminated with a non-printing character called a line feed character.
5. In the space provided below explain the difference in ASCII text files found on Unix/Linux systems and Windows systems.

   line breaks are done with both a line feed and a carriage return in windows (\n and the funky p) whereas in Unix/Linux, it's just done with a line feed (\n)

6. Fundamental operations of all types of files include:
   1. opening a file.
   2. reading from a file.
   3. writing to a file.
   4. closing a file.
7. In Python when a file is successfully opened a file object is created that provides the method needed to access the file.
8. T All files must be opened before they can be read or written to.
9. Which built in function is used to open a text file for processing:
   A. fileopen
   B. openfile
   C. readfile
   D. open
   E. None of the above.
10. When accessing a file the *open* function requires two arguments. The first is the file the second is the permission needed.
11. Choose the mode by which we open a text file in order to add lines to the file:
   A. 'r'
   B. 'w'
   C. 'a'
   D. 'add'
   E. 'addline'
   F. None of the above.
12. T If a file is opened for writing the contents of the existing file will be overwritten and lost.
13. F The *readline* method on a file object returns a line of the text file after removing the end-of-line character.
14. T The second argument to an *open* function is optional when opening a file for reading.
15. When opening a file for access errors may occur these are called IOError and they cause *exceptions* to be thrown.
16. The code below is extracted from Figure 5-7 in your text. The questions which are shown below are related to this listing.

17. Lines are read and written to ASCII files as which Python data type: file object.

18. string processing refers to the operations on strings that allow them to be accessed, analyzed and updated.

19. Strings are collections therefore they can be traversed using a for loop.

20. T The elements of a string must be accessed using an index.

21. T Strings are immutable objects.

22. T All string operations that modify a string return a new string that is a modified version of the original string.

23. Given a string, *s = 'A sample string'* you will find below a series of operations applicable to strings that reference the string. In the space provided write the expected result:
    (a) *len(s)* result: 15.
    (b) *min(s)* result: .
    (c) *max(s)* result: t.
    (d) *'z' in s* result: False.
    (e) *'sa' in s* result: True.
    (f) *s.count(s)* result: 1.
    (g) *s.index('p')* result: 5.
    (h) *s[3:7]* result: ampl.
    (i) *s[4:]* result: mple string
    (j) *s[:5]* result: A sam
    (k) *s.isalpha()* result: False.
    (l) *s.isupper()* result: False.
    (m) *s.lower()* result: a sample string
    (n) *s.upper()* result: A SAMPLE STRING
    (o) *s.split()* result: ['A', 'sample', 'string']

24. Given the string h = ' string? ' what is returned by the following call to a function on the h object: h.strip(' ?'). string

25. There are 3 basic types of errors when developing computer programs. Fill in the blanks below for each description:
    (a) syntax errors Errors in the code itself caught by the interpreter.
    (b) algorithmic errors Errors which despite correct compilation and execution result in the wrong answer.
    (c) exceptions Errors which occur during execution.

26. When a Python program is executing errors which occur create a special type of object referred to as an __exception__.

27. These objects must be either __caught__ or __handled__ to keep the program from crashing.

28. Python creates several different standard exceptions in response to errors which occur as the program executes. Below some of these are listed. Below each explain the circumstances under which they are created:

    (a) `ImportError`

    raised when an import statement fails

    (b) `IOError`

    raised when an input/output operation fails

    (c) `TypeError`

    raised when a local or global name is not found

    (d) `ValueError`

    raised when a built-in operation or function is applied to an appropriate type but an inappropriate value

    (e) `IndexError`

    raised when a sequence index is out of range

29. F In order to process *exceptions* we must import the exceptions module.

30. F If an exception is raised while executing a function the program is interrupted at that point and terminated.

31. An __try-except structure__ is a programming structure that redirects execution in the event that an exception is throw while executing codes found in its __try suite__, execution is re-directed to the part of the structure called the __exception handler__.

32. F A *try-except* structure may have multiple try-suites.

33. T A *try-except* structure may have multiple except-suites.

34. Examine the code segment shown below and answer the questions below:

```
1 # Part A: What it the structure below called?
2 try:
3     # Part B: What is this part of the structure called?
4     input_file = open(file_name, 'r')
```

```
5       input_file_opened = True

7       line = input_file.readline()

9       while line != empty_file:
10          print(line.strip('\n'))
11          line = input_file.readline()

13  exception:
14     #Part C: What is this part of the structure called?
15     print('File Open Error\n')
16     file_name = input("Enter file name: ")

18  # Part D: What type of exception will be raised by the block of code
19  # following the "try" portion of the above structure?
20  # Part E: If an exception is raised in the structure above
21  # the program will terminate immediately.
```

(a) The structure above is called an try-except structure

(b) Part B of the structure above is called the try suite

(c) Part C of the structure above is called the exception handler

(d) An IOError exception is raised.

(e) F If an exception is raised in the structure above the program will terminate immediately.

35. T IOError exceptions raised as a result of a file open error can be caught and handled.