**Files and Exceptions**

* Introduction

9.1 Q1: Which of the following statements is false?

a. Variables, lists, tuples, dictionaries, sets, arrays, pandas Series and pandas DataFrames offer long-term data storage.

b. Computers store files on secondary storage devices, including solid-state drives, hard disks and more.

c. Some popular text file formats are plain text, JSON (JavaScript Object Notation) and CSV (comma-separated values).

d. All of the above statements are true?

Answer: a. Variables, lists, tuples, dictionaries, sets, arrays, pandas Series and pandas DataFrames offer long-term data storage. Actually, variables, lists, tuples, dictionaries, sets, arrays, pandas Series and pandas DataFrames offer only *temporary* data storage. The data is lost when a local variable “goes out of scope” or when the program terminates.

9.1 Q2: Which of the following statements are false?

a. You can use JSON to serialize and deserialize objects to facilitate saving those objects to secondary storage and transmitting them over the Internet.

b. There are security vulnerabilities to serializing and deserializing data with the Python Standard Library’s pickle module.

c. The with statement automatically releases a resource at the end of the statement’s suite.

d. All of the above statements are true.

Answer: d. All of the above statements are true.

* Files

9.2 Q1: Which of the following statements is false?

a. Python views text files and binary files (for images, videos and more) as sequences of bytes.

b. As in lists and arrays, the first character in a text file and byte in a binary file is located at position 0, so in a file of n characters or bytes, the highest position number is n – 1.

c. For each file you open, Python creates a file object that you’ll use to interact with the file.

d. All of the above statements are true.

Answer: a. Python views text files and binary files (for images, videos and more) as sequences of bytes. Actually, Python views a text file as a sequence of characters and a binary file (for images, videos and more) as a sequence of *bytes*.

9.2 Q2: When a Python program begins execution, it creates three standard file objects: sys.stdin—the standard input file object, sys.stdout—the standard output file object, and sys.stderr—the standard error file object. You must import the sys module if you need to refer to these objects explicitly in your code, but this is rare. Which of the following statements is false?

a. Though these are considered file objects, they do not read from or write to files by default.

b. The input function implicitly uses sys.stdin to get user input from the keyboard.

c. Function print implicitly outputs to sys.stdout, which appears in the command line.

d. Python implicitly outputs program errors and tracebacks to sys.stdout, which appears in the command line.

Answer: d. Python implicitly outputs program errors and tracebacks to sys.stdout, which appears in the command line. Actually, Python implicitly outputs program errors and tracebacks to *sys.stderr*, which also appears in the command line.

* Text-File Processing
* Writing to a Text File: Introducing the with Statement

9.3 Q1: Which of the following statements is false?

a. Python imposes a convenient record structure on files.

b. The following code creates an accounts.txt file and write five client records to the file. Generally, records in text files are stored one per line, so we end each record with a newline character:

with open('accounts.txt', mode='w') as accounts:  
 accounts.write('100 Jones 24.98\n')  
 accounts.write('200 Doe 345.67\n')  
 accounts.write('300 White 0.00\n')  
 accounts.write('400 Stone -42.16\n')  
 accounts.write('500 Rich 224.62\n')

c. You can also write to a file with print (which automatically outputs a \n), as in

print('100 Jones 24.98', file=accounts)

d. All of the above statements are true.

Answer: a. Python imposes a convenient record structure on files. Actually, Python imposes no structure on a file, so notions such as records do not exist natively in Python*.* Programmers must structure files to meet their applications’ requirements.

9.3 Q2: Which of the following statements is false?

a. Many applications acquire resources, such as files, network connections, database connections and more. You should release resources when your application terminates.

b. The with statement acquires a resource and assigns its corresponding object to a variable.

c. The with statement allows the application to use the resource via that variable.

d. The with statement calls the resource object’s close method to release the resource when program control reaches the end of the with statement’s suite.

Answer: a. Many applications acquire resources, such as files, network connections, database connections and more. You should release resources when your application terminates. Actually, you should release resources as soon as they’re no longer needed—this practice ensures that other applications can use the resources as soon as possible, improving overall system performance.

9.3 Q3: Which of the following statements are false?

a. The built-in open function opens a file and associates it with a file object.

b. The mode argument specifies the file-open mode, indicating whether to open a file for reading from the file, for writing to the file or both.

c. The mode 'w' opens the file for writing, creating the file if it does not exist. If the file already exists, opening it for writing causes any new data to be appended to the end of the file.

d. At the end of the with statement’s suite, the with statement implicitly calls the file object’s close method to close the file.

Answer: c. The mode 'w' opens the file for writing, creating the file if it does not exist. If the file already exists, opening it for writing causes any new data to be appended to the end of the file. Actually, the mode 'w' opens the file for writing, creating the file if it does not exist. Be careful—opening an existing file for writing deletes any data in the file.

* Reading Data from a Text File

9.3 Q4: If the contents of a file should not be modified, open the file for         —another example of the principle of least privilege. This prevents the program from accidentally modifying the file.

a. writing only

b. reading only

c. reading and writing

d. none of the above

Answer: b. reading only

9.3 Q5: Which of the following statements are false?

a. The file object’s readlines method also can be used to read an entire text file. The method returns each line as a string in a list of strings.

b. Calling readlines for a large file can be a time-consuming operation, which must complete before you can begin using the list of strings.

c. Using the file object in a for statement enables your program to process each text line as it’s read.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Updating Text Files

No questions.

* Serialization with JSON

9.5 Q1: Which of the following statements are false?

a. Many libraries you’ll use to interact with cloud-based services such as Twitter, IBM Watson and others communicate with your applications via JSON objects.

b. JSON (JavaScript Object Notation) is a data-interchange format readable only by computers and used to represent objects (such as dictionaries, lists and more) as collections of name–value pairs.

c. JSON can represent objects of custom classes.

d. JSON has become the preferred data format for transmitting objects across platforms. This is especially true for invoking cloud-based web services, which are functions and methods that you call over the Internet.

Answer: b. JSON (JavaScript Object Notation) is a data-interchange format readable only by computers and used to represent objects (such as dictionaries, lists and more) as collections of name–value pairs. Actually, JSON (JavaScript Object Notation) is a text-based, *human-and-computer-readable*, data-interchange format used to represent objects (such as dictionaries, lists and more) as collections of name–value pairs.

9.5 Q2: Which of the following statements is false?

a. JSON objects are similar to Python lists. Each JSON object contains a comma-separated list of property names and values, in curly braces. For example, the following JSON object might represent a client record:

{"account": 100, "name": "Jones", "balance": 24.98}

b. JSON also supports arrays which, like Python lists, are comma-separated values in square brackets. For example, the following is an acceptable JSON array of numbers:

[100, 200, 300]

c. Values in JSON objects and arrays can be strings in double quotes, numbers, JSON Boolean values (true or false), null (to represent no value, like None in Python), arrays and other JSON objects.

d. All of the above statements are true.

Answer: a. JSON objects are similar to Python lists. Each JSON object contains a comma-separated list of *property names* and *values*, in curly braces. For example, the following key–value pairs might represent a client record:

{"account": 100, "name": "Jones", "balance": 24.98}

Actually, JSON objects are similar to Python dictionaries.

9.5 Q3: The json module enables you to convert objects to JSON (JavaScript Object Notation) text format. This is known as \_\_\_\_\_\_\_\_ the data.

a. chunking

b. calibrating

c. serializing

d. registering

Answer: c. serializing

9.5 Q4: Which of the following statements is false?

a. The following code writes an object in JSON format to a file:

import json  
with open('accounts.json', 'w') as accounts:  
 json.dump(accounts\_dict, accounts)

b. The snippet in Part (a) opens the file accounts.json and uses the json module’s dump function to serialize the dictionary accounts\_dict into the file.

c. JSON delimits strings with double-quote characters.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

9.5 Q5: The json module’s          function reads the entire JSON contents of its file object argument and converts the JSON into a Python object. This is known as          the data.

a. read, serializing

b. load, serializing

c. load, deserializing

d. read, deserializing

Answer: c. load, deserializing

9.5 Q6: Which of the following statements is false?

a. The json module’s dump function returns a Python string representation of an object in JSON format.

b. Using dumps with load, you can read JSON from a file and display it in a nicely indented format—sometimes called “pretty printing” the JSON.

c. When the dumps function call includes the indent keyword argument, as in

with open('accounts.json', 'r') as accounts:  
 print(json.dumps(json.load(accounts), indent=4))

the string contains newline characters and indentation for pretty printing—you also can use indent with the dump function when writing to a file:

d. All of the above statements are true.

Answer: a. The json module’s dump function returns a Python string representation of an object in JSON format. Actually, the json module’s *dumps* function (dumps is short for “dump string”) returns a Python string representation of an object in JSON format.

* Focus on Security: **pickle** Serialization and Deserialization

9.6 Q1: Which of the following statements is false?

a. The Python Standard Library’s pickle module can serialize objects into the data formats of many popular programming languages.

b. Pickle has potential security issues.

c. If you write your own pickling and de-pickling code, pickling can be secure.

d. All of the above statements are true.

Answer: a. The Python Standard Library’s pickle module can serialize objects into the data formats of many popular programming languages. Actually, pickle is Python specific.

* Additional Notes Regarding Files

9.7 Q1: Which of the following statements is false?

a. The file open modes for writing and appending raise a FileNotFoundError if the file does not exist.

b. Each text-file mode has a corresponding binary-file mode specified with b, as in 'rb' or 'wb+'.

c. You’d use binary-file modes if you were reading or writing binary files, such as images, audio, video, compressed ZIP files and many other popular custom file formats.

d. All of the above statements are true.

Answer: a. The file open modes for writing and appending raise a FileNotFoundError if the file does not exist. Actually, the writing and appending modes create the file if it does not exist.

9.7 Q2: Which of the following statements is false?

a. For a text file, the read method returns a string containing the number of characters specified by the method’s integer argument. For a binary file, the method returns the specified number of bytes. If no argument is specified, the method returns the entire contents of the file.

b. The readline method returns one line of text as a string, including the newline character if there is one.

c. The readline method issues an EndOfFileError when it encounters the end of the file.

d. The writelines method receives a list of strings and writes its contents to a file.

Answer: c. The readline method issues an EndOfFileError when it encounters the end of the file. Actually, readline returns an empty string when it encounters the end of the file.

* Handling Exceptions

9.8 Q1: Various types of exceptions can occur when you work with files. Which of the following statements is false?

a. A FileNotFoundError occurs if you attempt to open a non-existent file for reading with the 'r' or 'r+' modes.

b. A PermissionsError occurs if you attempt an operation for which you do not have permission. This might occur if you try to open a file that your account is not allowed to access or create a file in a folder where your account does not have permission to write, such as where your computer’s operating system is stored.

c. A LockedError (with the error message 'I/O operation on closed file.') occurs when you attempt to write to a file that has already been closed.

d. All of the above statements are true.

Answer: c. A LockedError (with the error message 'I/O operation on closed file.') occurs when you attempt to write to a file that has already been closed. Actually, a ValueError (with the error message 'I/O operation on closed file.') occurs when you attempt to write to a file that has already been closed.

* Division by Zero and Invalid Input

9.8 Q2: Which of the following statements is false?

a. Attempting to divide by 0 results in a ZeroDivisionError.

b. When a program attempts to divide by zero, the interpreter is said to raise an exception of type ZeroDivisionError.

c. When an exception is raised in an interactive IPython session, it displays the exception’s traceback, then terminates the IPython session.

d. If an exception occurs in a script, IPython terminates the script and displays the exception’s traceback.

Answer: c. When an exception is raised in an interactive IPython session, it displays the exception’s traceback, then terminates the interactive IPython session. Actually, when an exception is raised in an interactive IPython session, it terminates the snippet, displays the exception’s traceback, then shows the next In [] prompt so you can input the next snippet.

9.8 Q3: The int function raises a          if you attempt to convert to an integer a string (like 'hello') that does not represent a number.

a. NameError

b. Conversion-Error

c. Value-Error

d. None of the above

Answer: c. Value-Error

* try Statements

9.8 Q4: The following code uses exception handling to catch and handle any ZeroDivisionErrors and ValueErrors that arise.

|  |
| --- |
| * while True: |
| * try: |
| * number1 = int(input('Enter numerator: ')) |
| * number2 = int(input('Enter denominator: ')) |
| * result = number1 / number2 |
| * except ValueError: |
| * print('You must enter two integers\n') |
| * except ZeroDivisionError: |
| * print('Attempted to divide by zero\n') |
| * else: |
| * print(f'{number1:.3f} / {number2:.3f} = {result:.3f}') |
| * break |

Where in the code could either or both of these errors arise?

a. line 3

b. line 5

c. lines 3 and 4

d. lines 3, 4 and 5

Answer: d. lines 3, 4 and 5

9.8 Q5: Which of the following statements is false?

a. A try clause may be followed by one or more except clauses that immediately follow the try clause’s suite.

b. The except clauses also are known as exception handlers.

c. Each except clause specifies the type of exception it handles.

d. After the last except clause, an optional else clause specifies code that should execute only if the code in the try suite raised an exception.

Answer: d. After the last except clause, an optional else clause specifies code that should execute only if the code in the try suite raised an exception. Actually, after the last except clause, an optional else clause specifies code that should execute only if the code in the try suite *did not* raise exceptions.

9.8 Q6: Which of the following statements is false?

a. When an exception occurs in a try suite, it terminates immediately.

b. When there are except handlers, program control transfers to the first one that matches the type of the raised exception. If there are no except handlers, a process called stack unwinding occurs.

c. When an except clause successfully handles an exception, program execution resumes with the finally clause (if there is one), then with the next statement after the try statement.

d. After an exception is handled, program control returns to the raise point.

Answer: d. After an exception is handled, program control returns to the raise point. Actually, after an exception is handled, program control *does* *not* return to the raise point—rather, control resumes after the try statement.

* Catching Multiple Exceptions in One except Clause

9.8 Q7: Which of the following statements is false?

a. A try clause is often followed by several except clauses to handle various types of exceptions.

b. When no exceptions occur in the try suite, program execution resumes with the else clause (if there is one); otherwise, program execution resumes with the next statement after the try statement.

c. If several except suites are identical, you can catch those exception types by specifying them as a tuple in a single except handler, as in:

except [type1, type2, …] as variable\_name

d. You can use the variable name in the optional as-clause to reference the exception object in the except suite.

Answer: c. If several except suites are identical, you can catch those exception types by specifying them as a list in a single except handler, as in:

except [*type1*, *type2*, *…*] as *variable\_name*

Actually, c. If several except suites are identical, you can catch those exception types by specifying them as a *tuple* in a single except handler, as in:

except (*type1*, *type2*, *…*) as *variable\_name*

* What Exceptions Does a Function or Method Raise?

No questions.

* What Code Should Be Placed in a try Suite?

9.8 Q8: Which of the following statements is false?

a. Wrap a separate try statement around every statement that raises an exception.

b. For proper exception-handling granularity, each try statement should enclose a section of code small enough that, when an exception occurs, the specific context is known and the except handlers can process the exception properly.

c. If many statements in a try suite raise the same exception types, multiple try statements may be required to determine each exception’s context.

d. All of the above statements are true.

Answer: a. Wrap a separate try statement around every statement that raises an exception. Actually, place in a try suite a significant logical section of a program in which several statements can raise exceptions, *rather than* wrapping a separate try statement around every statement that raises an exception.

* **finally** Clause

9.9 Q1: Which of the following statements is false?

a. Operating systems typically can prevent more than one program from manipulating a file at once.

b. When a program finishes processing a file, the program should close it to release the resource. This enables other programs to use the file (if they’re allowed to access it).

c. Closing a file creates a resource leak in which the file resource is not available to other programs.

d. All of the above statements are true.

Answer: c. Closing a file creates a resource leak in which the file resource is not available to other programs. Actually, closing a file helps *prevent* a resource leak in which the file resource is not available to other programs because a program using the file never closes it.

9.9 Q2: Which of the following statements is false?

a. A try statement may have a finally clause as its last clause after any except clauses or else clause.

b. The finally clause is guaranteed to execute, regardless of whether its try suite executes successfully or an exception occurs.

c. In other languages that have finally, this makes the finally suite an ideal location to place resource-deallocation code for resources acquired in the corresponding try suite.

d. In Python, we prefer the else statement for resource-deallocation code and place other kinds of “clean up” code in the finally suite.

Answer: d. In Python, we prefer the else statement for resource-deallocation code and place other kinds of “clean up” code in the finally suite. Actually, in Python, we prefer the *with* *statement* for resource-deallocation code and place other kinds of “clean up” code in the finally suite.

9.9 Q3: Which of the following statements is false?

a. When the finally clause terminates, program control continues with the next statement after the try statement. In an IPython session, the next In [] prompt appears.

b. The following code includes a try statement in which an exception occurs in the try suite:

try:  
 print('try suite that raises an exception')  
 int('hello')  
 print('this will not execute')  
except ValueError:  
 print('a ValueError occurred')  
else:  
 print('else will not execute because an exception occurred')  
finally:   
 print('finally always executes')

c. When the finally clause terminates, program control continues with the next statement after the try statement. In an IPython session, the next In [] prompt appears.

d. The finally clause executes only when an exception occurs in the corresponding try suite.

Answer: d. The finally clause executes only when an exception occurs in the corresponding try suite. Actually, the finally clause *always* executes, *regardless* of whether an exception occurs in the corresponding try suite.

9.9 Q4: Which of the following statements is false?

a. Most resources that require explicit release, such as files, network connections and database connections, have potential exceptions associated with processing those resources.

b. A program that processes a file might raise IOErrors. For this reason, robust file-processing code normally appears in a try suite containing a with statement to guarantee that the resource gets released.

c. When code is in a try suite, you can catch in except handlers any exceptions that occur and you do not need a finally clause because the with statement handles resource deallocation.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Explicitly Raising an Exception

9.10 Q1: Which of the following statements is false?

a. The simplest form of the raise statement is

raise ExceptionClassName

b. In most cases, when you need to raise an exception, it’s recommended that you customize the exception type with a meaningful exception name.

c. The raise statement creates an object of the specified exception class. Optionally, the exception class name may be followed by parentheses containing arguments to initialize the exception object—typically to provide a custom error message string.

d. Code that raises an exception first should release any resources acquired before the exception occurred.

Answer: b. In most cases, when you need to raise an exception, it’s recommended that you customize the exception type with a meaningful exception name. Actually, in most cases, when you need to raise an exception, it’s recommended that you use one of Python’s many built-in exception types.

* (Optional) Stack Unwinding and Tracebacks

9.11 Q1: Which of the following statements is false?

a. Each exception object stores information indicating the precise series of function calls that led to the exception. This is helpful when debugging your code.

b. In the following function definitions, function1 calls function2 and function2 raises an Exception:

def function1():  
 function2()

def function2():  
 raise Exception('An exception occurred')

c. In an interactive IPython session, calling function1 in Part (b) results in the following traceback:

--------------------------------------------------------------------  
Exception Traceback (most recent call last)  
<ipython-input-3-c0b3cafe2087> in <module>()  
----> 1 function1()  
  
<ipython-input-1-a9f4faeeeb0c> in function1()  
 1 def function1():  
----> 2 function2()  
 3   
  
<ipython-input-2-c65e19d6b45b> in function2()  
 1 def function2():  
----> 2 raise Exception('An exception occurred')  
  
Exception: An exception occurred

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

9.11 Q2: Which of the following statements is false?

a. A traceback shows the type of exception that occurred followed by the complete function-call stack that led to the raise point. The stack’s          function call is listed          and the          is         , so the interpreter displays the following text as a reminder:

Traceback (most recent call last)

a. bottom, first, top, last.

b. bottom, last, top, first

c. top, first, bottom, last

d. top, last, bottom, first

Answer: a. bottom, first, top, last.

9.11 Q3: Which of the following statements is false?

a. Raising an exception in a finally suite can lead to subtle, hard-to-find problems.

b. If an exception occurs and is not processed by the time the finally suite executes, stack unwinding occurs.

c. If the finally suite raises a new exception that the suite does not catch, program execution terminates.

d. For this reason, a finally suite should always enclose in a try statement any code that may raise an exception, so that the exceptions will be processed within that suite.

Answer: c. If the finally suite raises a *new* exception that the suite does not catch, program execution terminates. Actually, if the finally suite raises a new exception that the suite does not catch, the first exception is lost, and the new exception is passed to the next enclosing try statement.

* Intro to Data Science: Working with CSV Files
* Python Standard Library Module csv

9.12 Q1: Which of the following statements is false?

a. The csv module provides functions for working with CSV files. Many other Python libraries also have built-in CSV support.

b. The csv module’s documentation recommends opening CSV files with the additional keyword argument newline='' to ensure that newlines are processed properly

c. The csv module’s writer function returns an object that writes CSV data to the specified file object.

d. CSV files cannot contain spaces after commas.

Answer: d. CSV files cannot contain spaces after commas. Actually, CSV files generally do not contain spaces after commas, but some people use them to enhance readability.

9.12 Q2: Which of the following would be accepted by a program strictly looking for four integers of data in CSV format?

a. 100, 85 77,9

b. 100,85,,77,9

c. '100,85,77,9’='

d. 100,85, 77,9

Answer: d. 100,85, 77,9

* Reading CSV Files into Pandas DataFrames

9.12 Q3: Which of the following statements is false?

a. You can load a CSV dataset into a DataFrame with the pandas function read\_csv.

b. The following IPython session loads and displays the CSV file accounts.csv:

In [1]: import pandas as pd  
  
In [2]: df = pd.read\_csv('accounts.csv',   
 ...: names=['account', 'name', 'balance'])  
 ...:  
  
In [3]: df  
Out[3]:   
 account name balance  
0 100 Jones 24.98  
1 200 Doe 345.67  
2 300 White 0.00  
3 400 Stone -42.16  
4 500 Rich 224.62

c. The names keyword argument specifies the DataFrame’s column names. If you do not supply the names keyword argument, read\_csv assumes that the CSV file’s first row is a comma-delimited list of column names.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Reading the Titanic Disaster Dataset

9.12 Q4: Which of the following statements about DataFrames and the Titanic dataset is false?

a. For large datasets, displaying the DataFrame shows only the first 30 rows, followed by “…” and the last 30 rows.

b. To save space, we can view the first five and last five rows with DataFrame methods head and tail. Both methods return five rows by default, but you can specify the number of rows to display as an argument.

c. Pandas adjusts each column’s width, based on the widest value in the column.

d The value in the age column of row 1305 is NaN (not a number), indicating a missing value in the dataset.

Answer: c. Pandas adjusts each column’s width, based on the widest data value in the column. Actually, Pandas adjusts each column’s width, based on the widest value in the column *or* based on the column name, whichever is wider.

* Simple Data Analysis with the Titanic Disaster Dataset

No questions.

* Titanic Passenger Age Histogram

9.12 Q5: Which of the following statements is false?

a. When you call describe on a DataFrame containing both numeric and non-numeric columns, describe calculates the statistics below only for the numeric columns—in the Titanic dataset, for example, just the age column:

age  
count 1046.00  
mean 29.88  
std 14.41  
min 0.17  
25% 21.00  
50% 28.00  
75% 39.00  
max 80.00

b. When performing calculations, Pandas ignores missing data (NaN) by default.

c. For non-numeric data, describe displays different descriptive statistics, such as unique (the number of unique values in the result), top (the most frequently occurring value in the result) and freq (the number of occurrences of the top value).

d. A DataFrame’s hist method automatically analyzes each column’s data, and produces a corresponding histogram for each.

Answer: d. A DataFrame’s hist method automatically analyzes each column’s data, and produces a corresponding histogram for each. Actually, a DataFrame’s hist method automatically analyzes only each *numerical* column’s data, and produces a corresponding histogram for each.