Name: ________________________ **Class:** ___________________ **Date:** __________ **ID: A**

CST-186 Chapter 7 Study Guide

True/False

Indicate whether the statement is true or false.

____ 1. Using a text file, you can easily store complex objects.

____ 2. If you don’t specify the number of characters to be read when invoking the text file method `read()`, Python returns the entire file as a string.

____ 3. Any file that's open can be written to.

____ 4. A closed file cannot be read from or written to.

____ 5. Python lets you pickle a variety of objects, including numbers, strings, tuples, lists, and dictionaries.

____ 6. The `shelve` module provides objects that allow random access to pickled objects.

____ 7. An unhandled exception halts a program.

____ 8. A `try` statement prevents exceptions from occurring.

____ 9. An `except` clause tests whether or not an exception has occurred.

____ 10. The `else` clause block of a `try` statement executes only if no exception is raised in the `try` statement.

Multiple Choice

Identify the choice that best completes the statement or answers the question.

____ 11. What text file method reads a specified number of characters from a text file and returns them as a string?

a. `read()`
b. `readline()`
c. `readlines()`
d. `readfile()`

____ 12. What text file method reads a specified number of characters from the current line in a text file and returns them as a string?

a. `read()`
b. `readline()`
c. `readlines()`
d. `readfile()`

____ 13. What text file method reads all of the lines in a text file and returns them as elements in a list?

a. `read()`
b. `readline()`
c. `readlines()`
d. `readfile()`

____ 14. What text file method writes a single string to a text file?

a. `write()`
b. `writeline()`
c. `writelines()`
d. `writefile()`

____ 15. What text file method writes a list of strings to a text file?

a. `write()`
b. `writeline()`
c. `writelines()`
d. `writefile()`

____ 16. Which text file access mode allows only file reading?

a. `"r"`
b. `"r+"`
c. `"read"`
d. `"read+"`

____ 17. What will the value of the variable `data` be after the following code executes? (Assume that file.txt is a valid text file that can be read by the code)

```
f = open("file.txt", "r")
data = f.read()
f.close()
```

a. The first character of the text file
b. The first line of the text file
c. The entire text file as a string
d. The entire text file as a list of strings

____ 18. What will the value of the variable `data` be after the following code executes? (Assume that file.txt is a valid text file that can be read by the code)

```
f = open("file.txt", "r")
data = f.readlines()
f.close()
```

a. The first character of the text file
b. The first line of the text file
c. The entire text file as a string
d. The entire text file as a list of strings

____ 19. What will be displayed by the following code? (Assume that file.txt is a valid text file that can be read by the code)

```
f = open("file.txt", "r")
for x in f:
    print x
f.close()
```

a. The first character of the text file
b. The first line of the text file
c. The entire text file
d. None of these

____ 20. In the following code, what is `x`? (Assume the code is a valid call to `cPickle.dump()`)

```
cPickle.dump(x, y)
```

a. data to be pickled
b. a file object to be written to
c. a file access mode
d. an index number

____ 21. In the following code, what is `y`? (Assume the code is a valid call to `cPickle.dump()`)

```
cPickle.dump(x, y)
```

a. data to be pickled
b. a file object to be written to
c. a file access mode
d. an index number

____ 22. How can pickled objects written to a file, using `cPickle.dump()` function, be accessed?

a. sequentially
b. randomly
c. they cannot be accessed
d. None of these

____ 23. How can objects written to a shelf, using the `shelve` module, be accessed?

a. sequentially
b. randomly
c. they cannot be accessed
d. None of these

____ 24. What will be displayed by the following code?

```
try:
    num = float("ten")
except:
    print "Exception!",

print "End."
```

a. `Exception!`
b. `End`
c. `Exception! End.`
d. an exception raised message

____ 25. What will be displayed by the following code?

```
try:
    num = float("10")
except:
    print "Exception!",

print "End."
```

a. `Exception!`
b. `End.`
c. `Exception! End`
d. an exception raised message

____ 26. What will be displayed by the following code?

```
try:
    num = float("ten")
except:
    print "Exception!",

print num
print "End."
```

a. `Exception!`
b. `End.`
c. `Exception! End`
d. an exception raised message

____ 27. What will be displayed by the following code?

```
try:
    print float("ten")
except(TypeError):
    print "TypeError"
except(ValueError):
    print "ValueError"
```

a. `10.0`
b. `TypeError`
c. `ValueError`
d. an exception raised message

____ 28. What will be displayed by the following code?

```
try:
    print float(None)
except(TypeError):
    print "TypeError"
except(ValueError):
    print "ValueError"
```

a. `10.0`
b. `TypeError`
c. `ValueError`
d. an exception raised message

____ 29. In the following what will the variable `e` receive?

```
try:
    num = float("ten")
except(ValueError), e:
    print "Exception!", e
```

a. the exception’s argument
b. the exception's parameter
c. the exception's variable
d. the exception’s name

____ 30. What will be displayed by the following code?

```
try:
    num = float("10")
except:
    print "Exception!",
else:
    print num,

print "End."
```

a. `End.`
b. `Exception! End.`
c. `10.0 End.`
d. an exception raised message

Completion

Complete each statement.

31. Text files are ____________________, meaning that they can be used without modification on different operating systems.

32. Before you can read from (or write to) an existing text file, you need to ____________________ it.

33. Whenever you’re done with a file, it’s good programming practice to ____________________ it.

34. The ____________________ module allows you to pickle and store more complex data in a file.

35. The ____________________ module allows you to store and randomly access pickled objects in a file.

Matching

Match each item with a statement below

a. Exception
b. Exception argument
c. Pickle
d. Plain text file
e. Shelf
f. Trap
g. `dump()`
h. `sync()`
i. `try` statement
j. `except` clause

____ 36. An error that occurs during the execution of a program.

____ 37. To catch an exception.

____ 38. A value associated with an exception that has been raised.

____ 39. Code that sections off statements that could potentially raise an exception.

____ 40. An object written to a file that acts like a dictionary, providing random access to a group of pickled objects.

____ 41. A file that's made up of only ASCII characters.

____ 42. A `cPickle` function that writes a pickled version of an object to a file.

____ 43. To store complex objects in files.

____ 44. Code that contains statements that are executed only if an exception is raised.

____ 45. A `shelf` method that forces changes to be written to a file.

Short Answer

46. What kind of information are text files good for storing?

47. What kind of information is best stored using the `cPickle` module?

48. When you write an object to a shelf file, is it written to the disk immediately?

49. When should you trap for exceptions?

50. Why should you trap for specific exception types?