CS303E Week 12 Worksheet: Files

Name: _____ EID: _____

Read the questions carefully, and answer each question in the space provided. Use scratch paper to do your work and then copy your answers neatly and legibly onto the test paper. Only answers recorded on the test paper will be graded.

1. (10 points: 1 point each) The following are true/false questions. **Write either T or F in the boxes at the bottom of page 1.** If there's any counterexample, it's false.

   (a) Using the `open()` function will result in an error when providing the name of a file that does not exist on your computer.

   (b) Closing a file using `.close()` is important because it ensures that system resources are released and any data remaining in memory buffers is written.

   (c) When reading from a file using `.readline()`, the newline character at the end of each line is included in the returned string.

   (d) Directories (folders on Windows) themselves do not have file paths, because file paths refer only to the location of files within directories.

   (e) The `.close()` method must be called on files in the same order that they were opened to avoid potential issues with file handling.

   (f) When you `write()` to a file in Python, the new data is appended at the end of the original contents of the file.

   (g) In Python, using `open()` on a directory allows you to read or write to all files within that directory.

   (h) When using the `"w"` mode to write to a file, only strings can be written.

   (i) The `.readlines()` method will always return all lines from a file, regardless of whether the file handle has previously read from the file.

   (j) Files are used to store data persistently, while directories are used to provide a hierarchical arrangement of files.

a	b	c	d	e	f	g	h	i	j

Page total: _____/10

Questions 2-7 are multiple choice. Each counts 2 points. **Write the letter of the BEST answer in the box on page 3. Please write your answer in UPPERCASE. Each problem has a single answer.**

2. When reading data from a file, which of the following methods is considered the most memory-efficient for processing large files?

   A. `.read()`

   B. `.readline()`

   C. `.readlines()`

   D. All three methods have similar memory efficiency.

3. What is the difference between a relative file path and an absolute file path?

   A. Relative paths always start from the directory where Python is installed, while absolute paths can start from any directory.

   B. Relative paths cannot be used for files in our current working directory, while absolute paths can be used for any file.

   C. Relative paths specify a file's location based on the current working directory, while absolute paths specify a file's location starting from the top-level (root) directory of the file system.

   D. Relative paths are typically used for specifying input files, whereas absolute paths are exclusively reserved for specifying output files.

4. Which of the following approaches correctly prints the entire contents of a file, with leading and trailing whitespace removed from each line?

   A.
   ```
   content = file.read().strip()
   print(content)
   ```

   B.
   ```
   line = file.readline().strip()
   while line:
       print(line)
       line = file.readline().strip()
   ```

   C.
   ```
   lines = file.readlines()
   for line in lines:
       line = line.strip()
   print(lines)
   ```

   D. All of the above

   E. B and C

5. Assume a list of strings is passed to the following function:

```
def friendshipIsMagic(twilight):
    fluttershy = open("rainbowdash.txt", "w")
    rarity = []
    for applejack in twilight:
        for spike in applejack:
            if spike not in rarity:
                rarity.append(spike)
    fluttershy.write(str(rarity))
    fluttershy.close()
```

What does the code primarily do?

A. Creates a new text file named `"rainbowdash.txt"` and writes to it strings from the list `twilight`.
B. Writes unique characters found in each string in the list `twilight` to the existing content of `"rainbowdash.txt"`.
C. Prints the lists of unique characters found in the file `"rainbowdash.txt"`
D. Creates a new text file named `"rainbowdash.txt"` and writes to it a list containing the unique characters found in the list `twilight`.

6. Which of the following are differences between `print()` and `.write()` in Python?

A. `.write()` is for writing to files, while `print()` is only for console output.
B. `.write()` only accepts strings, while `print()` can handle various data types.
C. by default `print()` automatically appends a newline character at the end, while `.write()` does not.
D. `print()` accepts multiple parameters, while `.write()` only accepts one.
E. All of the above.
F. A and C

7. If you only have a file open in `"a"` mode, what happens if you attempt to read from the file?

A. The file is read without any issue.
B. An error is raised.
C. The file is read, but only the new content added in `"a"` mode is retrieved.
D. Reading the file in `"a"` mode results in an empty string.

2	3	4	5	6	7

Page total: _____/14

The following questions (2 points each) ask you to trace the behavior of some Python code and identify the output of that code. For each question, write the output for the code segment in the provided box. Assume the following two files are downloaded, and that any possible changes made to them do not persist between questions.

`hungergames.txt`

```
katniss peeta primrose
finnick effie haymitch
joanna cinna rue
```

`barney.txt`

```
1 2 buckle my shoe
3 4 shut the door
5 6 pick up sticks
```

8.
```
muppets = open("hungergames.txt", "r")
gonzo = muppets.readline()
kermit = []
while gonzo:
    msPiggy = gonzo.split(" ")
    fozzie = msPiggy[-1]
    kermit.append(fozzie)
    gonzo = muppets.readline()
muppets.close()
print(kermit)
```

9.
```
ghostface = open("hungergames.txt", "r")
myers = ghostface.readlines()
for krueger in range(len(myers)):
    chucky = myers[krueger].split()
    if krueger == 0:
        for friday13 in chucky:
            print(friday13, end = " ")
    print(chucky[0], end = " ")
ghostface.close()
```

Page total: _____/4

10.
```
monstersINC = open("barney.txt", "w")
monstersINC.write("BOO!")
monstersINC.close()
sully = open("barney.txt")
mike = sully.readline()
sully.close()
print(mike)
```

11.
```
stevenConnie = open("hungergames.txt", "a")
stevenConnie.write("\ncrystalgems\n")
stevenConnie.close()
garnet = open("hungergames.txt", "r")
pearl = garnet.read().split()
garnet.close()
amethyst = pearl[-1]
print(amethyst)
```

12.
```
peridot = open("hungergames.txt", "r")
lapis = open("hungergames.txt", "r")
peridot.readline()
peridot.readline()
lapis.readline()
print(lapis.readline().strip())
lapis.close()
peridot.close()
```

13.
```
peterpan = open("barney.txt")
tinkerbell = peterpan.readline()
peterpan.close()
peterpan.close()
print(tinkerbell)
```

14.
```
plankton = open("barney.txt")
karen = plankton.read()
plankton.close()
mrsPuff = [0] * 10
for larry in karen.split():
    pearl = len(larry)
    mrsPuff[pearl] += 1
print(mrsPuff)
```

Page total: _____/4