

## Assignment 18 Solutions

### Q1. Describe the differences between text and binary files in a single paragraph.

**Ans:** The differences between Text Files and Binary Files are:

**Text files** are special subset of binary files that are used to store human readable characters as a rich text document or plain text document. Text files also store data in sequential bytes but bits in text file represents characters. where as **Binary files** are those typical files that store data in the form of sequence of bytes grouped into eight bits or sometimes sixteen bits. These bits represent custom data and such files can store multiple types of data (images, audio, text, etc) under a single file.

### Q2. What are some scenarios where using text files will be the better option? When would you like to use binary files instead of text files? ¶

**Ans:** text files are commonly used for storage of information. They avoid some of the problems encountered with other file formats, such as endianness, padding bytes, or differences in the number of bytes in a machine word.

Binary files are used to store data more compactly. In the text file, a special character whose ASCII value is 26 inserted after the last character to mark the end of file. In the binary file no such character is present. Files keep track of the end of the file from the number of characters present.

### Q3. What are some of the issues with using binary operations to read and write a Python integer directly to disc?

**Ans:** When we read or write a python integer using binary operations

1. Binary operations deal with raw data
2. One needs to identify how many bytes one would read or write.

### Q4. Describe a benefit of using the with keyword instead of explicitly opening a file ?

**Ans:** When a file is opened using the **with** keyword, it automatically closes the file if some exceptions occur after opening a file, or at the end of the file . Thereby not leaving a file in open mode and also there would no need to explicitly close a file. Using **with** means that the file will be closed as soon as you leave the block. This is beneficial because closing a file is something that can easily be forgotten and ties up resources that you no longer need.

### Q5. Does Python have the trailing newline while reading a line of text? Does Python append a newline when you write a line of text?

**Ans:** Yes, Python have the trailing newline while reading a line of text. When we write a newline has to be provided in python explicitly.

### Q6. What file operations enable for random-access operation?

**Ans:** The file operations enable for random-access operation are `seek()` and `tell()`

### Q7. When do you think you'll use the struct package the most?

**Ans:** The `struct` package is mostly used while converting a common python types into `C` language types.

### Q8. When is pickling the best option?

**Ans:** Pickling is best option for creating a new binary file using python.

### Q9. When will it be best to use the shelve package?

**Ans:** `Shelve` package is used to pickle data but treats the entire file as dictionary. A "shelf" is a persistent, dictionary-like object. The difference with "dbm" databases is that the values (not the keys) in a shelf can be essentially arbitrary Python objects — anything that the pickle module can handle. This includes most class instances, recursive data types, and objects containing lots of shared sub-objects. The keys are ordinary strings.

Syntax: `shelve.open(filename, flag='c', protocol=None, writeback=False)`

### Q10. What is a special restriction when using the shelve package, as opposed to using other data dictionaries?

**Ans:** Only string data type can be used as key in this special dictionary (shelve package) object, whereas any pickle Python object can be used as value.

In [ ]:

1

