**Python Advanced Assignment 18**

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

Ans-) Text files and binary files are two types of files that store data in different formats. Text files store data in human-readable format, with each line of text terminated by a newline character. Binary files, on the other hand, store data in a non-human-readable format, with data represented as binary code. Binary files may include special characters that are not easily represented in text format, such as images, videos, and executable programs.

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 typically used when data needs to be easily readable and editable by humans or other programs. Examples of scenarios where text files would be the better option include storing configuration data, log files, and source code. Binary files, on the other hand, are used for storing non-textual data, such as images, videos, and audio files. They are also used for storing executable code and data that requires efficient storage and retrieval.

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

directly to disc?

Ans-) One issue with using binary operations to read and write Python integers directly to disk is that the byte order of the integer may differ depending on the platform on which it was written. Another issue is that the size of the integer may differ depending on the platform. To overcome these issues, it is recommended to use a standardized format, such as the struct module, when writing integers to disk.

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

Ans-) Using the with keyword to open a file in Python provides a number of benefits over explicitly opening a file. The with statement ensures that the file is properly closed after use, even if an exception is raised during file processing. It also provides a cleaner and more readable way to write file processing code.

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-) When reading a line of text in Python, the trailing newline character is included in the string that is returned. When writing a line of text, Python does not automatically append a newline character, unless specified in the write() method.

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

Ans-) The file operations that enable random-access operation include seek() and tell(). The seek() method allows you to move the file pointer to a specific location in the file, while the tell() method returns the current position of the file pointer.

Q7. When do you think you&#39;ll use the struct package the most?

Ans-) The struct package is used for converting between Python values and binary data. It is particularly useful when working with low-level I/O operations, such as when reading and writing data to hardware devices or network sockets.

Q8. When is pickling the best option?

Ans-) Pickling is the best option when you need to serialize Python objects and store them in a way that can be easily loaded and accessed later. It is particularly useful for storing complex data structures, such as lists, dictionaries, and objects.

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

Ans-) The shelve package is best used when you need to store large amounts of data that need to be accessed quickly. It is particularly useful for storing and retrieving large dictionaries, lists, and objects.

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

dictionaries?

Ans-) One special restriction when using the shelve package is that it only supports string keys. This means that you cannot use integer, float, or other non-string keys to access the data in a shelve object.