**Assignment: (Core Java)**

**14 : File Handling**

**Que.1 Introduction to File I/O in Java (java.io package)**

**Ans.1** In Java, File I/O (Input and Output) refers to reading data from files and writing data to files.The java.io package provides a wide range of classes and interfaces to handle file operations, streams, and buffers.

**I/O in Java?**

* Input → Data coming into the program e.g., reading from a file, keyboard, network.
* Output → Data going out of the program e.g., writing to a file, console, network.

**The java.io Package:** The java.io package provides:

* Streams for handling data:
  + Byte Streams → read/write binary data (images, audio, etc.).
  + Character Streams → read/write text data (Unicode supported).
* File Handling classes → work directly with files and directories.

**Que.2 FileReader and FileWriter Classes**

**Ans.2 FileReader Class**

* Purpose: To read text data (characters) from files.
* Belongs to java.io package.
* It is a subclass of InputStreamReader.

**FileWriter Class**

* Purpose: To write text data (characters) into files.
* Belongs to java.io package.
* It is a subclass of OutputStreamWriter.

**Que.3 BufferedReader and BufferedWriter**

**Ans.3 1. BufferedReader Class**

* Purpose: Reads text efficiently from a file (or any character input stream) using buffering.
* Internally keeps a buffer (default size 8192 characters) to reduce disk read operations.
* Subclass of Reader.

**2. BufferedWriter Class**

* Purpose: Writes text efficiently to a file (or any character output stream) using buffering.
* Stores data in a buffer and writes to the file in chunks, reducing disk access.
* Subclass of Writer.

**Que.4 Serialization and Deserialization**

**Ans.4** Serialization and Deserialization are mechanisms used to convert objects into a stream of bytes and vice versa. This is useful for:

* Saving objects to files
* Sending objects over networks
* Storing objects in databases

These are part of the java.io package.

**Serialization**: Serialization is the process of converting a Java object into a byte stream so that it can be saved to a file, sent over a network, or stored in a database.

* Purpose: Preserve the object’s state for later use.
* Requirements: The class of the object must implement the Serializable interface.

.

**Deserialization:** Deserialization is the process of converting a byte stream back into a copy of the original Java object. Restores the object with its original state.