**Core Java - Assignment**

**Module - 1**

1. **Java Input/Output (I/O) :**

**Theory : -**

1. Streams in Java (InputStream, OutputStream):

-> InputStream : InputStream is an abstract class that represents an input stream of bytes. It is the superclass of all classes representing an input stream of bytes. Some commonly used subclasses of InputStream include FileInputStream, ByteArrayInputStream , and BufferedInputStream .

Key Methods

-> read() : Reads the next byte of data from the input stream.

-> read(byte[] b) : Reads some number of bytes from the inputstream and stores them into the buffer array b.

-> read(byte[] b, int off, int len) : Reads upto len bytes data from the input stream into an array of bytes.

-> close() : Closes the input stream and releases any system resources associated with it.

-> OutputStream : OutputStream is an abstract class that represents an output stream of bytes. It is the superclass of all classes representing an output stream of bytes. Some commonly used subclasses of OutputStream include FileOutputStream, ByteArrayOutputStream , and BufferedOutputStream.

Key Methods

-> write(int b) : Writes the specified byte to the output stream.

-> write(byte[] b) : Writes b. length bytes from the specified byte array to the output stream.

-> write(byte[] b, int off, int len) : Writes len bytes from the specified byte array starting at offset off to the output stream.

-> close() : Closes the output stream and releases any system resources associated with it.

1. Reading and Writing Data Using Streams

-> In Java, streams are used to perform input and output (I/O) operations. The java.io package provides a wide range of classes for handling I/O operations, including InputStream and OutputStream for byte-based I/O, and Reader and Writer for character-based I/O.

-> Byte-Based I/O

-> InputStream and OutputStream : InputStream and OutputStream are abstract classes that represent input and output streams of bytes, respectively.

-> Character-Based I/O

-> Reader and Writer : Reader and Writer are abstract classes that represent input and output streams of characters, respectively.

-> Buffered Streams

-> Buffered streams are used to improve the efficiency of I/O operations by reducing the number of I/O operations performed. They buffer the input and output, allowing for larger chunks of data to be read or written at once.

1. Handling File I/O Operations

-> Handling file I/O operations in Java involves reading from and writing to files. The java. io package provides a comprehensive set of classes for performing these operations. Here, we'll cover the basics of handling file I/O operations, including reading from and writing to files using both byte-based and character-based streams.

-> Byte-Based I/O

->Reading from a File using FileInputStream : The FileInputStream class is used to read raw bytes from a file.

-> Writing to a File using FileOutputStream : The FileOutputStream class is used to write raw bytes to a file.

-> Character-Based I/O

-> Reading from a File using FileReader : The FileReader class is used to read characters from a file.

-> Writing to a File using FileWriter : The FileWriter class is used to write characters to a file.

-> Buffered Streams :Buffered streams are used to improve the efficiency of 1/0 operations by reducing the number of 1/0 operations performed. They buffer the input and output, allowing for larger chunks of data to be read or written at once.

-> Reading from a File using BufferedReader : The BufferedReader class is used to read text from an input stream efficiently.

-> Writing to a File using BufferedWriter : The BufferedWriter class is used to write text to an output stream efficiently.