Java I/O Cheat Sheet javaconceptoftheday.com What is stream? A stream is a sequence of data generated by the input source and consumed by the output destination. There are two types of I/O streams in Java. Byte Streams Character Streams 1) Byte Streams ✓ Byte streams read and write data byte by byte i.e. 8 bits maximum at a time. ✓ These streams are most suitable to process the binary files like image files, audio files, video files, executable files etc. ✓ All byte stream classes in Java are of type InputStream and OutputStream. ✓ InputStream classes are used to read data from a source and OutputStream classes are used to write data to a destination. 2) Character Streams ✓ Character streams read and write data character by character. ✓ Character streams are mainly used to process text files. All character stream classes in Java are of type Reader and Writer. Reader classes are used to read source files and Writer classes are used to write destination files. Byte Streams Byte Streams Read Write It is the super class for all the classes which It is the super class for all the classes which InputStream OutputStream read data from a source in bytes. write data to a destination in bytes. It is used to write binary files like image It is used to read binary files like image FileOutputStream FileInputStream files, audio files, video files etc. files, audio files, video files etc. These classes provide filter to other These classes provide filter to other input FilterOutputStream FilterInputStream output streams. streams. It provides buffer to an input It provides buffer to an output BufferedInputStream BufferedOutputStream stream. stream. It is used to read primitive data It is used to write primitive data DataInputStream DataOutputStream from an input stream in a into an output stream in a machine independent way. machine independent way. It additionally maintains It additionally maintains CheckedInputStream checksum of the data being read. CheckedOutputStream checksum of the data being written. It provides cipher which can be It provides cipher which can be used to decrypt the data before used to encrypt the data before CipherOutputStream CipherInputStream data is written. data is read. It is used to write compressed It is used to read compressed data from an input stream in the deflate data to an output stream in the DeflaterOutputStream DeflaterInputStream deflate compression format. compression format. It is used to decompress the data It is used to decompress the data stored in a deflate compression stored in a deflate compression InflaterOutputStream InflaterInputStream format before writing to an output format before reading from an input stream. stream. An input stream which updates An output stream which updates DigestInputStream associated message digest while associated message digest while DigestOutputStream reading. writing. It adds functionality to other output PrintStream streams so that they can print different data representations more conveniently. It internally uses a byte array to read It internally uses a byte array to ByteArrayInputStream ByteArrayOutputStream write data to a destination. data from a source. It is used to read data which is It is used to write Java objects into an output stream. It is called previously written by ObjectInputStream ObjectOutputStream serialization. ObjectOutputStream. It is called deserialization. PipedInputStream and PipedOutputStream are connected to each other to form a PipedInputStream communication pipe. Data is read from a PipedInputStream by one thread and data is PipedOutputStream written to the corresponding PipedOutputStream by some other thread. It reads data from multiple streams SequenceInputStream sequentially one by one. **Character Streams** Character Streams Write Read It is the super class for all the classes which are It is the super class for all the classes which are Reader Writer used to write characters into destination files. used to read characters from source files. It acts as a bridge from byte streams to It acts a bridge from character streams to character streams. It reads bytes and byte streams. It encodes characters into InputStreamWriter InputStreamReader bytes using specified charset before writing decodes them into characters using specified charset. into destination file. It is used to write character files. It is used to read character files. FileReader FileWriter It provides buffer to read character files fast It provides buffer to write character files fast BufferedWriter BufferedReader and efficiently. and efficiently. It is used to keep track of line LineNumberReader numbers. It provides character buffer array to read It provides character buffer array to write CharArrayReader CharArrayWriter character files. into character files. It is used to read filtered character streams. It is used to write filtered character streams. FilterReader FilterWriter It allows characters to be pushed back PushBackReader into the stream. PipedReader and PipedWriter are connected to each other to form a pipe of communication. Character PipedWriter PipedReader data is read through PipedReader by one thread and data is written into corresponding PipedWriter by another thread. It is used to read data from strings. Here, It is used to write data into string buffer, StringReader strings are the sources of data. StringWriter which is then can be used to construct a string. It is used to write formatted text to an output PrintWriter stream.