# CSCI 2120:
## Software Design & Development II

*UNIT3: I/O management*

*io api*

**PushbackInputStream**

# Overview

1. Introduction
2. PushbackInputStream class declaration
3. PushbackInputStream constructors
4. PushbackInputStream methods
5. PushbackInputStream Examples

# Introduction

- **PushbackInputStream in Java** is an input stream that pushes (i.e. returns) a single byte or character back into the input stream to reread.

- In other words, `PushbackInputStream` is a buffer stream that adds “push back” functionality to an input stream to reread “unread” bytes (or push back read bytes). That’s why it is also called pushback buffer.

- It is used on an input stream to allow a byte to be read and then returned to the stream.

- For example, suppose we have an input stream that contains a string of bytes “Love”. If we read two bytes, we would have read Lo.

- If we call `unread()` method of `PushbackInputStream` to push back the last byte that we have read, the subsequent read will return ‘o’ and the next reads will read ve.

# PushbackInputStream class declaration

`PushbackInputStream` is a concrete subclass of `FilterInputStream` that extends `InputStream` superclass. It also implements `Closeable`, and `AutoCloseable` interfaces.

The general syntax to declare `PushbackInputStream` class in Java is as follows:

```
public class PushbackInputStream
       extends FilterInputStream
       implements Closeable, AutoCloseable
```

`PushbackInputStream` class was added in Java 1.0 version. It is present in `java.io.PushbackInputStream` package.

# PushbackInputStream Constructors

`PushbackInputStream` class defines the following constructors in Java. They are as follows:

**1. PushbackInputStream(InputStream inputStream):**
This form of constructor creates a PushbackInputStream object that allows one byte to be returned to the input stream.

**2. PushbackInputStream(InputStream inputStream, int numBytes):**
This form of constructor creates a stream that has a pushback buffer that is numBytes long. It allows multiple bytes to be returned to the input stream.

# PushbackInputStream Methods

In addition to methods inherited from `FilterInputStream` and `InputStream` classes, `PushbackInputStream` also defines several important methods. They are as follows:

# PushbackInputStream Methods

| Method | Description |
|---|---|
| int available() | This method returns an actual number of bytes that can be read (or skipped over) from the underlying input stream without blocking by the next invocation of a method.<br><br>If this input stream is closed by invoking its close() method, or an I/O error occurs, this method returns an IOException. |
| void close() | This method closes the underlying input stream and releases any system resources associated with the stream. The close() method will throw an IOException if an I/O error occurs. |
| void mark(int readlimit) | This method marks the current position in this input stream. |
| boolean markSupported() | This method examines if this input stream supports the mark and reset methods, which it does not. |

# PushbackInputStream Methods

| Method | Description |
|---|---|
| int read() | This method reads the next byte of data from the underlying input stream. It returns the next byte of data, or -1 if the end of the stream has been reached.<br><br>If this input stream is closed by invoking its close() method, or an I/O error occurs, this method returns an IOException. |
| int read(byte[ ] b, int n, int m) | It reads up to m bytes of data from this input stream into an array of bytes. |
| long skip(long n) | This method skips over and discards n bytes of data from the underlying input stream. |

# PushbackInputStream Methods

| Method | Description |
|---|---|
| void unread(byte[ ] b) | This method pushes back an array of bytes b by copying it to the front of the pushback buffer. |
| void unread(byte[ ] b, int n, int m) | It pushes back m bytes of an array of bytes b, starting from nth byte, by copying it to the front of the pushback buffer. |
| void unread(int b) | This method pushes back a byte by copying it to the front of the pushback buffer. |

**Note:**
All the above three `unread()` methods throw an `IOException` if there is not sufficient room in the pushback buffer (i.e. pushback buffer is full) for the specified byte, or this input stream has been closed by calling its `close()` method.

# Example 1:  Read data from File

1. Let's take an example program where we will understand how to use the `PushbackInputStream` to unread bytes to the input stream and reread them.

In this example, we will read `"I Love Java Programming"` from the `"myfile.txt"`. Look at the source code to understand better.

# Example 1:  Read data from File

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.PushbackInputStream;

public class PushbackInputStreamTester1 {
   public static void main(String[] args) throws IOException {
       String filepath = "./src/myfile.txt";
       FileInputStream fis = new FileInputStream(filepath);
       PushbackInputStream pushback = new PushbackInputStream(fis);

       // Read a single byte at a time and print it.
       byte bytedata;
       while((bytedata = (byte)pushback.read()) != -1) {
           System.out.print( (char) bytedata );

           // Unread the last byte that we have just read.
           pushback.unread(bytedata);

           // Reread the byte we unread (or pushed back).
           bytedata = (byte) pushback.read();
           System.out.print( (char) bytedata );
       }
   }
}
```

# Example 1:  Read data from File

**Output:**

```
WWeellccoommee  ttoo  UUNNOO  CCoommppuutteerr  SScciieennccee
```

As you can observe in the output, the program reads each byte from the file twice.

For example, “Welcome” is read as “WWeellccoommee”.

# END