Compilers

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

- Java PushbackReader

Today's Lecture

- **PushbackReader** - A character-stream reader that allows characters to be pushed back into the stream.
- For example, assume you are reading the following data:
  abcdef
- Reading will start from the first character in the stream.
  abcdef
  ↑   **The next char that is**
  Next   **read will be a**
  char

- After doing a read the a is consumed.
  bcdef
  ↑   **The next char that is**
  Next   **read will be b**
  char

- You are allowed to unread a character (put a char back into a stream). Here is the stream after unreading the a:
  abcdef
  ↑   **The next char that is**
  Next   **read will be a**
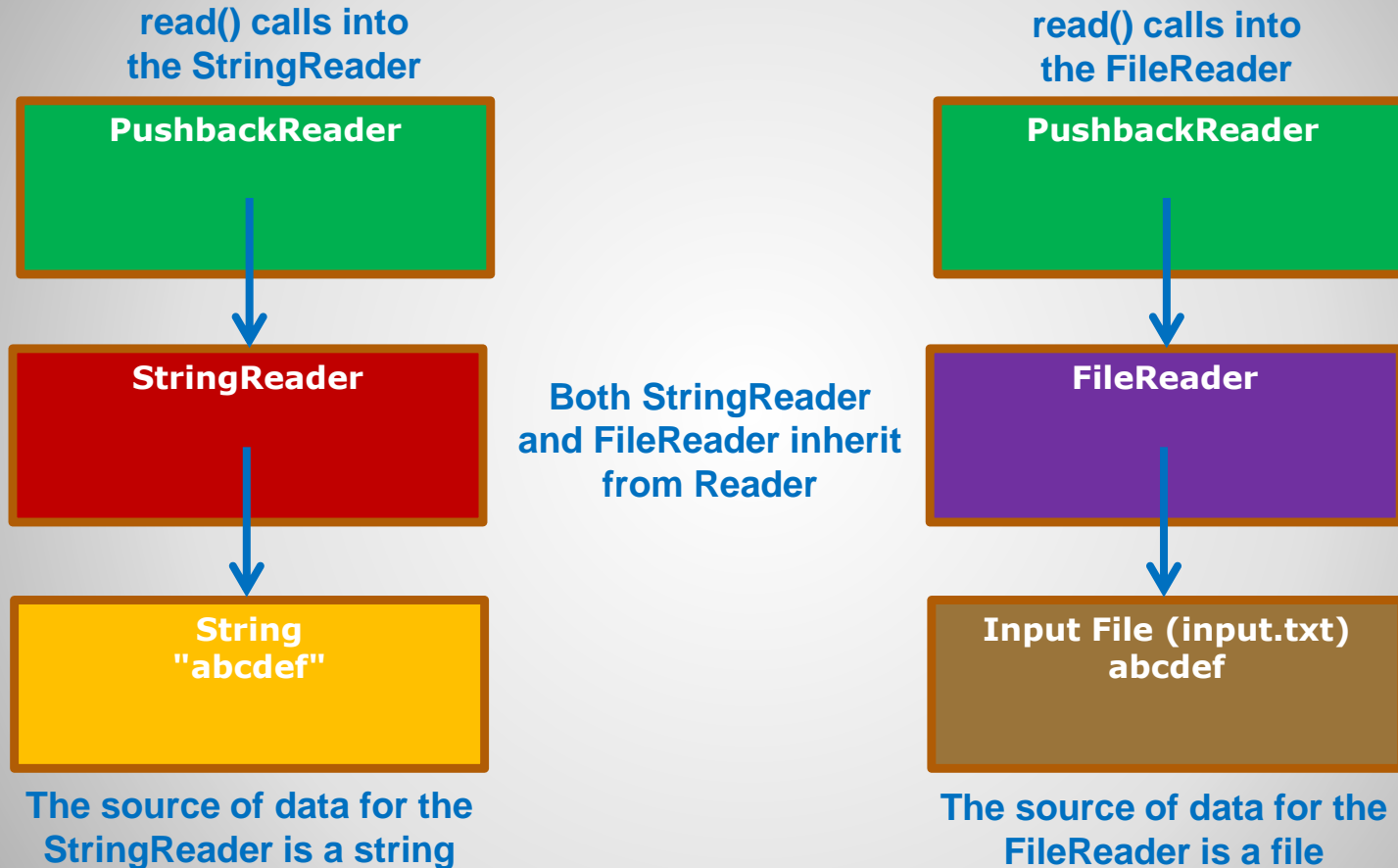 **char**

PushbackReader

PushbackReader Methods

- **PushbackReader(Reader in)** – Constructor that takes a Reader. Creates a new pushback reader with a one-character pushback buffer.

- **int read()** – Reads a single character from the stream. The char is returned as an integer value. This integer value is the ASCII character code. Check www.asciitable.com for all ASCII code numbers.

- **void unread(int c)** – Pushes back a single character by copying it to the front of the pushback buffer.

PushbackReader Methods

PushbackReader and Data Sources

- PushbackReader's relationship with its data source.

read() calls into the StringReader

read() calls into the FileReader

| PushbackReader | PushbackReader |

Both StringReader and FileReader inherit from Reader

| StringReader | FileReader |

| String "abcdef" | Input File (input.txt) abcdef |

The source of data for the StringReader is a string

The source of data for the FileReader is a file

PushbackReader and Data Sources

Create PushbackReader Instance that uses a StringReader

- The PushbackReader constructor takes a Reader as a parameter.
- Reader is an abstract class for reading character streams.
- The StringReader class is derived from Reader. StringReader is a character stream whose source is a string.

```
try {
    String data = "abc";

    StringReader strReader = new StringReader(data);

    PushbackReader pbReader = new PushbackReader(strReader);

    // Code to use PushbackReader goes here

} catch (IOException e) {
    throw new RuntimeException(e);
}
```

Source of data for StringReader

StringReader uses the data variable as its data source

The PushbackReader uses the StringReader as its data source

Create PushbackReader Instance from StringReader

<u>**Create PushbackReader Instance that uses a FileReader**</u>

- The PushbackReader constructor takes a Reader as a parameter.
- Reader is an abstract class for reading character streams.
- The FileReader class is derived from Reader. FileReader is uses to read character data from a file.

```
try {
    FileReader fileReader = new FileReader("input.txt");

    PushbackReader pbReader = new PushbackReader(fileReader);

    // Code to use PushbackReader goes here

} catch (FileNotFoundException e) {
    throw new RuntimeException(e);
} catch (IOException e) {
    throw new RuntimeException(e);
}
```

The file "input.txt" is the source of data for the FileReader

The PushbackReader uses the FileReader as its data source

Create PushbackReader Instance from FileReader

Reading and Unreading Characters

- The code below reads a character from the stream and then pushes it back into the stream.
- When the second read is done it returns the character that was pushed back into the stream.

```
try {
    // pbReader setup code goes here

    int c;
    c = pbReader.read();                    ← Reads first char in the stream
    System.out.println((char) c);

                                            Push the char that was read
                                            back into the stream (will be
    pbReader.unread(c);                   ← put at the start of the stream)

    c = pbReader.read();                  ←
    System.out.println((char) c);           Reads the original first char in
                                            the stream again since it was
                                            put back into the stream

} catch (IOException e) {
    throw new RuntimeException(e);
}
```

Reading and Unreading Characters

PushbackReader and End of File

- To detect that the end of file (EOF) has been reached you need to check for the end of file character.

- If the int value returned from read() is -1 then the end of file has been reached.

- IMPORTANT! Calling unread with -1 causes problems. Initially, when read is called at the EOF it will return -1. If that -1 is unread and read is called again after that it will return the value 65535 (instead of -1).

PushbackReader and End of File

- **End of Slides**

End of Slides