File Edit View Help

File Edit View Help

Programming Review

Part 05

File Edit View Help

Exceptions and File I/O

File Edit View Help

Exceptions

- Provides a means to alert and handle exceptional run-time events
- 3 Elements
  - –Creating Exceptions
  - –Using Exceptions
  - –Handling Exceptions
- Creating Exceptions
  - –“Extending” existing Exception
  - –Only create Constructors that sets the Exception Message
- Using Exceptions
  - –Method Definition uses “throws” to indicate which Exceptions may occur
  - –Use the reserved word “throw” followed by a constructor when the Exception occurs

Syntax for Creating an Exception

```
public class <<Exception identifier>> extends <<Existing Exception>>
{
        //Constructors
        public <<Exception identifier>>()
        {
                super(“<<Exception Message>>”);
        }
}
```

Example

```
public class DivideByZeroException extends Exception
{
    public DivideByZeroException()
    {
        super(“Divided by Zero Exception: Attempted to divide by Zero”);
    }
}
```

File Edit View Help

Exceptions

- Provides a means to alert and handle exceptional run-time events
- 3 Elements
  - –Creating Exceptions
  - –Using Exceptions
  - –Handling Exceptions
- Creating Exceptions
  - –“Extending” existing Exception
  - –Only create Constructors that sets the Exception Message
- Using Exceptions
  - –Method Definition uses “throws” to indicate which Exceptions may occur
  - –Use the reserved word “throw” followed by a constructor when the Exception occurs

Syntax for Using an Exception

```
<<Method declaration>> throws <<Exception>>, …
{
        …
        throw new <<Exception Constructor>>;
}
```

Example

```
public void calculateValue() throws DivideByZeroException, UnknownOpException
{
     …
     if(denominator == 0)
          throw new DivideByZeroException();
     …
     if(opNotRecognized())
          throw new UnknownOpException();
}
```

File Edit View Help

Exceptions

- Provides a means to alert and handle exceptional run-time events
- 3 Elements
  - –Creating Exceptions
  - –Using Exceptions
  - –Handling Exceptions
- Creating Exceptions
  - –“Extending” existing Exception
  - –Only create Constructors that sets the Exception Message
- Using Exceptions
  - –Method Definition uses “throws” to indicate which Exceptions may occur
  - –Use the reserved word “throw” followed by a constructor when the Exception occurs

Syntax for Handling an Exception

```
try
{
        <<Method(s) that throws exceptions>>
}
catch(<<Exception Type>> e)
{
        <<Handle the Exception>>
}
```

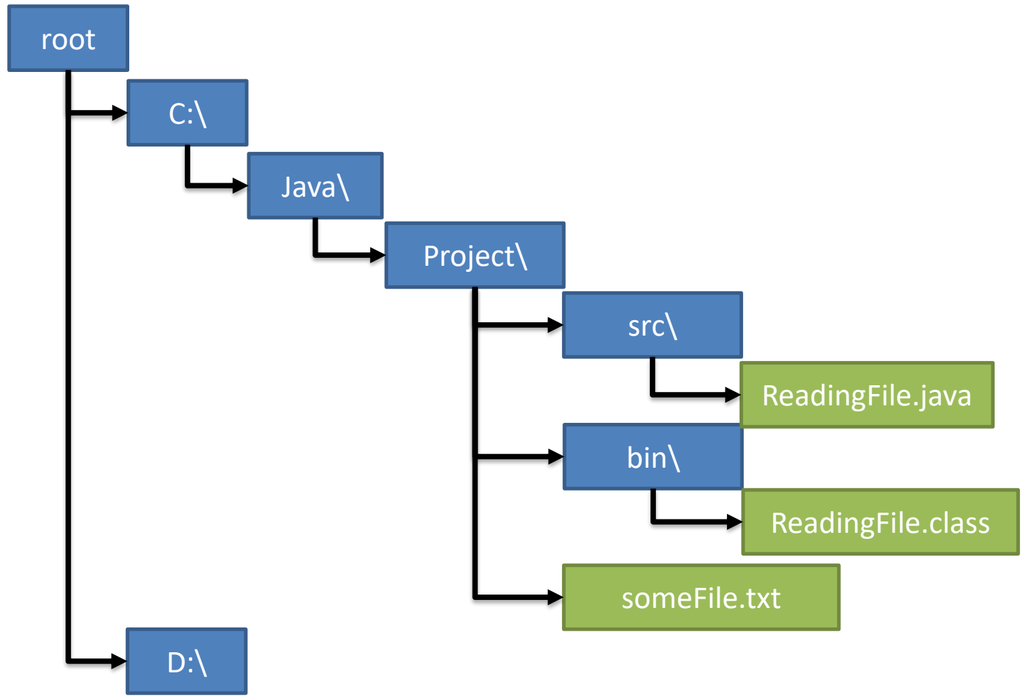
Example

```
try
{
      calculateValue();
}
catch(DivideByZeroException e)
{
      e.printStackTrace();
}
catch(UnknownOpException e)
{
      e.printStackTrace();
}
catch(Exception e)
{
      e.printStackTrace();
}
```

File I/O

- File Input and Output (I/O) allows a program to both read and write files to the secondary storage
- File Systems
  - –Organize Data in Secondary Storage
    - •Directory – Tree Structure
    - •Folders – Containers for Files
    - •Root is the starting point
  - –Address (File Path)
    - •Absolute (C:\Folder\Folder\...)
    - •Relative (.\Folder or ..\Folder)
- Java Project's Root Directory
  - –Contains a source ("SRC") folder
  - –Contains the bytecode ("BIN") folder
  - –Put files in a Project's Root Directory

File System Example

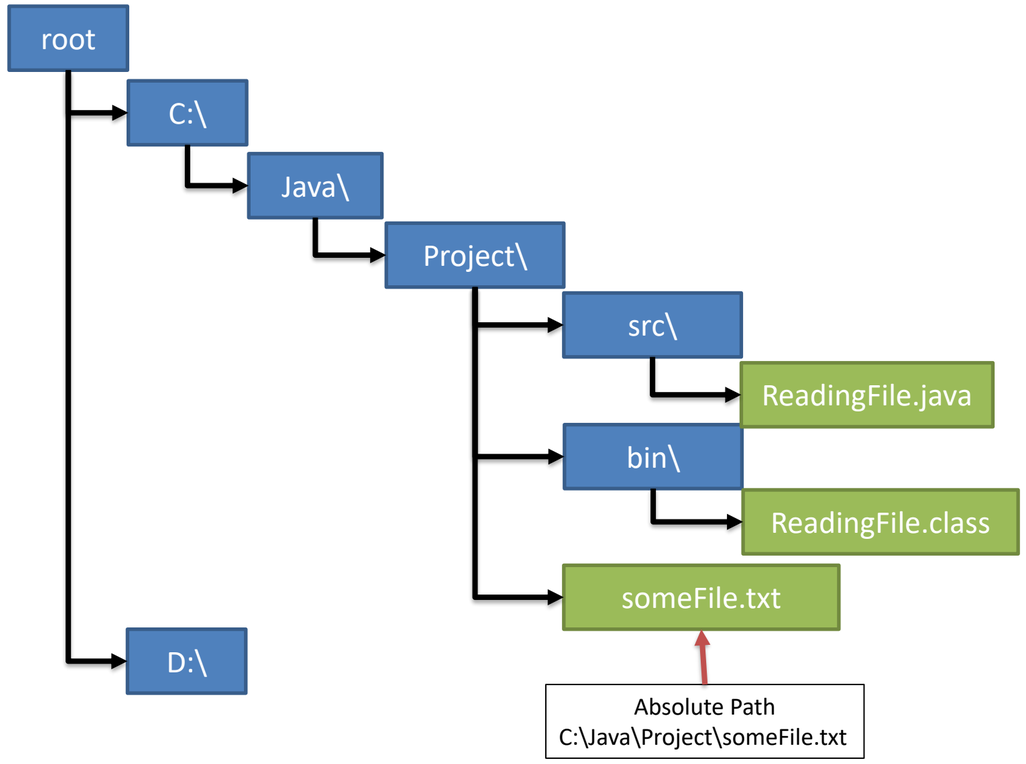


File Edit View Help

File I/O

- File Input and Output (I/O) allows a program to both read and write files to the secondary storage
- File Systems
  - –Organize Data in Secondary Storage
    - •Directory – Tree Structure
    - •Folders – Containers for Files
    - •Root is the starting point
  - –Address (File Path)
    - •Absolute (C:\Folder\Folder\...)
    - •Relative (.\Folder or ..\Folder)
- Java Project's Root Directory
  - –Contains a source ("SRC") folder
  - –Contains the bytecode ("BIN") folder
  - –Put files in a Project's Root Directory

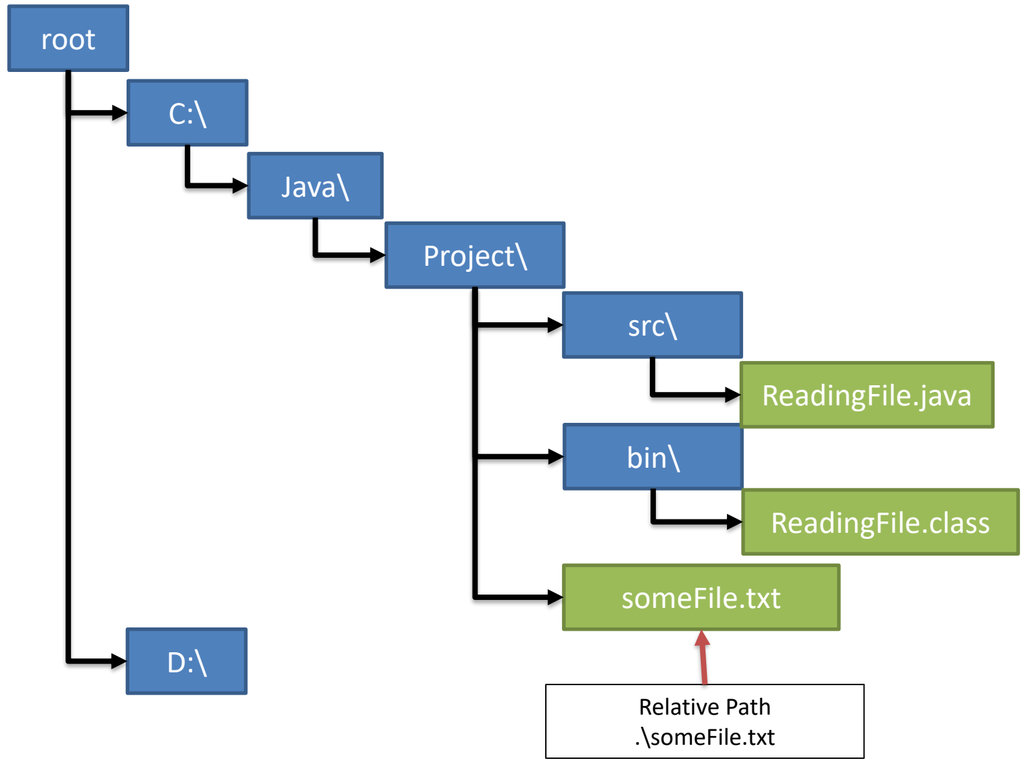
File System Example



File I/O

- File Input and Output (I/O) allows a program to both read and write files to the secondary storage
- File Systems
  - –Organize Data in Secondary Storage
    - •Directory – Tree Structure
    - •Folders – Containers for Files
    - •Root is the starting point
  - –Address (File Path)
    - •Absolute (C:\Folder\Folder\...)
    - •Relative (.\Folder or ..\Folder)
- Java Project's Root Directory
  - –Contains a source ("SRC") folder
  - –Contains the bytecode ("BIN") folder
  - –Put files in a Project's Root Directory

File System Example



File I/O

- File Format
  - How is information grouped
    - Header Data
    - Body Data
  - How information is Separated
    - Character / Byte length
    - Delimiters / Separators
- Plain Text Files
  - Readable Information
  - Only Files used in this Course
- Binary Files
  - Information stored in Bytes

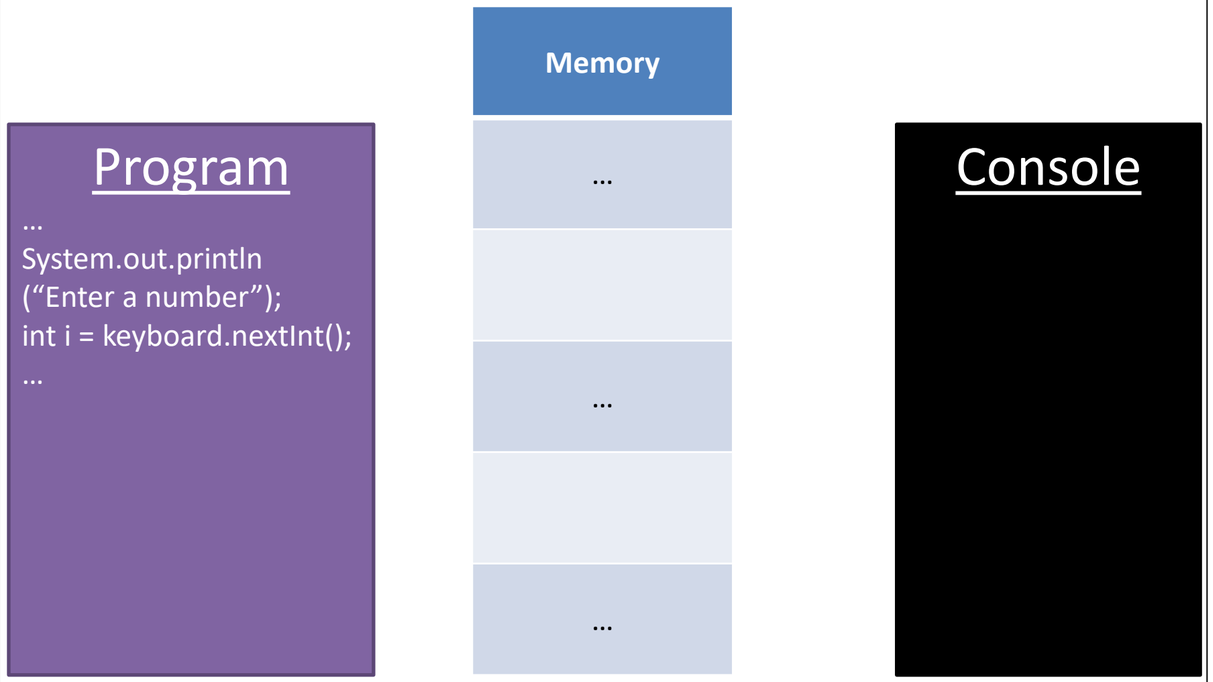
Common File Formats

- Plain Text
  - Readable words
  - Separated by Spaces (single space(s), tabs, end line)
  - <<word>><<space>…
- Tab Delimited
  - Information separated by Tabs (‘\t’) and end lines (‘\n’)
  - Spread Sheet
    - Rows are end lines
    - Columns are Tabs
  - <<information>>\t<<information>>\t…<<information>>\n
- Comma Separated
  - Information separated by Commas (‘,’) and end lines (‘\n’)
  - Spread Sheet
    - Rows are end lines
    - Columns are Tabs
  - <<information>>,<<information>>,…<<information>>\n

File I/O

- Streams and Buffers
  - A Stream is a Sequence of Data
  - Buffers provide space to temporarily hold information
  - Buffers holds Streamed information until "flushed"
  - Generally one direction
- System.out
  - Standard System output Stream
- System.in
  - Standard System input Stream

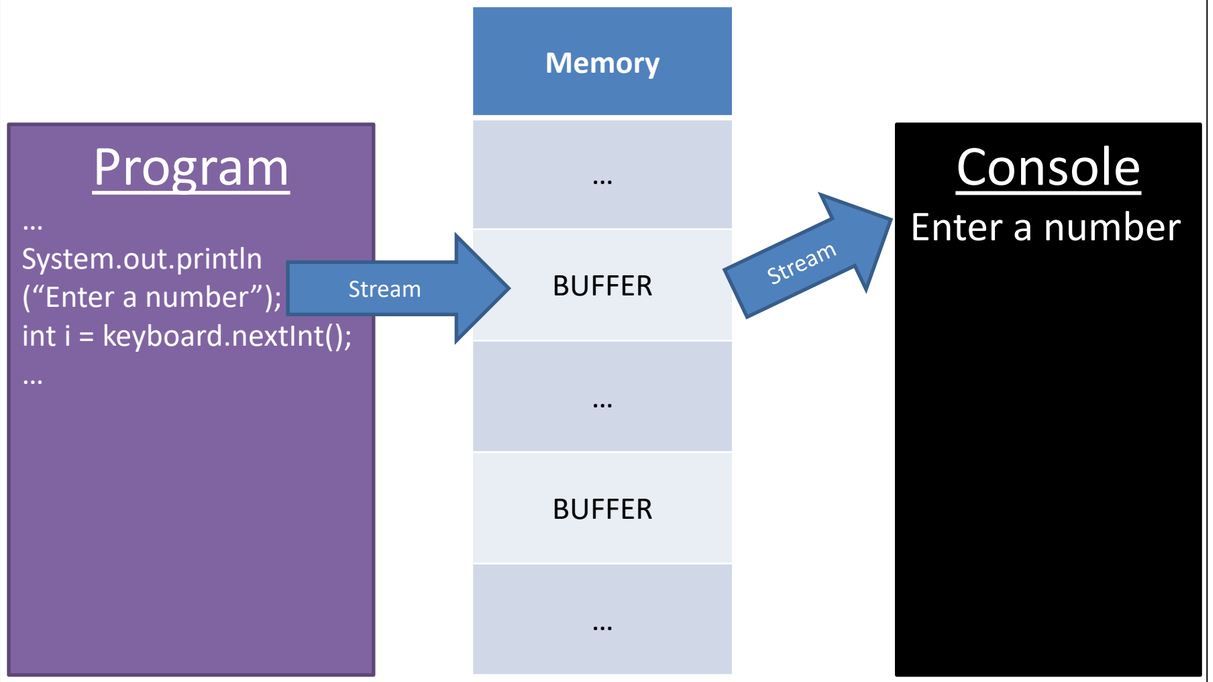
Stream and Buffer Example



File I/O

- Streams and Buffers
  - A Stream is a Sequence of Data
  - Buffers provide space to temporarily hold information
  - Buffers holds Streamed information until "flushed"
  - Generally one direction
- System.out
  - Standard System output Stream
- System.in
  - Standard System input Stream

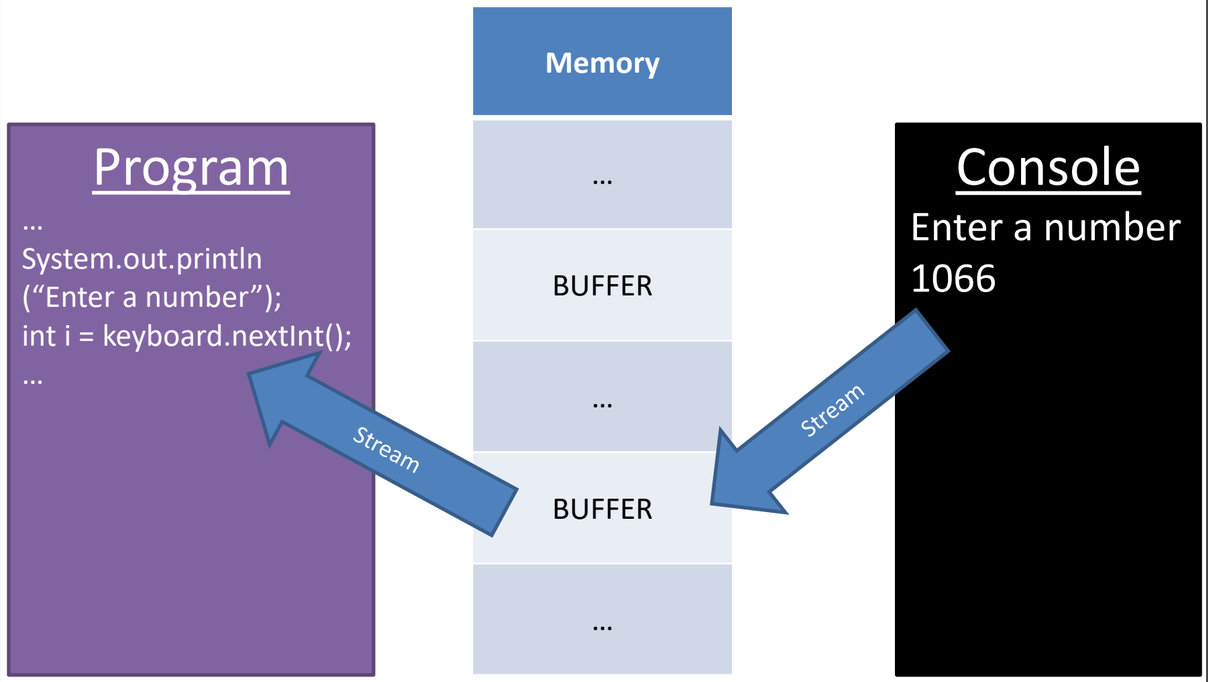
Stream and Buffer Example



File I/O

- Streams and Buffers
  - A Stream is a Sequence of Data
  - Buffers provide space to temporarily hold information
  - Buffers holds Streamed information until "flushed"
  - Generally one direction
- System.out
  - Standard System output Stream
- System.in
  - Standard System input Stream

Stream and Buffer Example



File Output

- PrintWriter
  - –Class that creates an object that can write to files
  - –Need to “import java.io.*” to use the type
  - –Construction and use need to be enclosed in a try-catch block
- Similar to System.out
  - –Streams information to a file
  - –Useful Methods
    - •print(<<String value>>);
    - •println(<<String value>>);
- Always CLOSE the file’s Stream
  - –Resource Leak
  - –Ensures all information has been saved (“flushed”)

Syntax

```
//Print Writer Construction to Create / Overwrite a file
PrintWriter <<pwID>> =
new PrintWriter(
new FileOutputStream(
new File(<<path+filename>>)));

//Print Writer Construction to Append to Existing file
PrintWriter <<pwID>> =
new PrintWriter(
new FileOutputStream(
new File(<<path+filename>>),true));

//Printing a new line
<<pwID>>.println(<<String value>>);

//Printing to same line
<<pwID>>.print(<<String value>>);

//Closing the PrintWriter Stream DO NOT FORGET THIS!
<<pwID>>.close();
```

File Input

- Scanner
  - –Scans any stream (File I/O, System.in, Network, Strings)
  - –Need to “import java.util.*” to use the type
  - –Construction and use, for files, need to be enclosed in a try-catch block
  - –All previous methods can be utilized
- Files are read left to right THEN top to bottom
  - –Just like the console you cannot go backwards
- Always CLOSE the file’s Stream
  - –Resource Leak
  - –Ensures all information has been saved (“flushed”)

Syntax

```
//Scanner construction to read a file
Scanner <<fsID>> = new Scanner(new
File(<<path+filename>>));

//Reading and storing an entire line (until the end line
‘\n’)
<<strVar>> = <<fsID>>.nextLine();

//Reading and storing the String until the next space
(any kind)
<<strVar>> = <<fsID>>.next();

//Reading and storing to the next integer encountered
<<intVar>> = <<fsID>>.nextInt();

//Close the Scanner DO NOT FORGET THIS
<<fsID>>.close();
```

Tips for Reading Files

- Use the type Scanner
- Make sure the file's path is correct
  - Use relative paths
  - Put files in Project's root directory
- Know the File Format
  - How is the information grouped
  - How is the information separated
- Files are read left to right then top to bottom
  - Cannot go backwards
- Always **CLOSE** the Scanner

- Reading a Plain Text File
  1. Open the file
  2. Read word by word (use method ".next()")
  3. Process the information
  4. Repeat Step 2 until the end of the file has been reached
- Reading a Tab Delimited
  1. Open the file
  2. Read an entire line (use method ".nextLine()")
  3. Split the line using Tabs
  4. Check if the information is valid
  5. Process the information
  6. Repeat Step 2 until the end of the file has been reached

File Edit View Help

Plain Text Example Concept

File Edit View Help

Plain Text Example

- Problem
  - Find the number of times a given word appears in a given file.

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Implementation

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Plain Text Example Detailed

```
…
while(fileScanner.hasNext())
{
   String next = fileScanner.next();
   …
}
```

File

SHREK

Written by

William Steig & Ted Elliott

SHREK
Once upon a time there was a lovely
princess. But she had an enchantment
upon her of a fearful sort which could
only be broken by love's first kiss.
She was locked away in a castle guarded
by a terrible fire-breathing dragon.
Many brave knights had attempted to
free her from this dreadful prison,
but non prevailed. She waited in the
dragon's keep in the highest room of
the tallest tower for her true love
and true love's first kiss. (laughs)
Like that's ever gonna happen. What
a load of - (toilet flush)

Allstar - by Smashmouth begins to play. Shrek goes about his
day. While in a nearby town, the villagers get together to go
after the ogre.

File Edit View Help

Programming Review

Part 05