Reading Text Files

InputStream and its subclasses read bytes. InputStreamReader reads input as characters. An InputStreamReader generally uses an InputStream (or subclass) as the source of bytes to use.

1. Use an InputStream and an InputStreamReader. This is the most flexible method since you can use *any* InputStream, including one that reads a file or URL. You can read 1 character at a time or read an array of chars (faster).

You must add try-catch to catch exceptions.

```
InputStream in = new FileInputStream(filename);
InputStreamReader reader = new InputStreamReader( in );
String result = "";
// read each character until you get -1, which means end-of-file
int c = in.read( );  // use a "while" loop to read chars
if ( c >= 0 )  result = result + (char)c;
// you can read into an array of chars (faster)
int size = 1024;
char[] chars = new char[size];
int count = reader.read(chars, 0, size);
count is the number of chars actually read. Don't assume that the array is full!

// when you get to the end, close the file.  Use a separate try-catch block.
if (reader != null ) reader.close( );
```

2. Use a FileReader. This is a convenience class for reading text files.

```
FileReader reader = new FileReader( filename );
```

the rest of the code is same as case 1.

3. Use a BufferedReader. BufferedReader can read a file as "lines" and create Strings.

It also requires using try - catch to catch IOException.

When you reach the end of the input, readLine() returns null.

```
FileReader reader = new FileReader( filename );
BufferedReader br = new BufferedReader( reader );
StringBuilder result = new StringBuilder();
String line;
// readLine() returns one line from file or null at end
while((line = br.readLine()) != null)
    result.append(line).append('\n');
// close the file - use try-catch here.
if ( br != null ) br.close();
```

4. Use a Scanner. Scanner is slower than BufferedReader but can convert input to many datatypes.

```
FileInputStream input = new FileInputStream("filename");
Scanner scanner = new Scanner( input );
```

```
// or use a File object
File file = new File("filename");
Scanner scanner = new Scanner( file );
```