

Section 13: File I/O

1) Files

- a) accessible on local machines via applications
- b) accessible on server via applet
- c) two types
 - stream
 - random

2) Stream file io

- a) input classes:
 - Reader
 - BufferedReader
 - InputStreamReader
 - FileReader
- b) output classes:
 - Writer
 - PrintWriter
 - FileWriter
- c) import java.io.*;
- d) reading lines of text
 - readline method of BufferedReader


```
String line;
BufferedReader inFile;
try {
    inFile = new BufferedReader( new FileReader("file.dat"));
    while ( (line = inFile.readLine() != null) {
        System.out.println(line);
    }
    inFile.close();
}
catch (IOException e) {
    System.err.println("Error in file! " + e.toString());
    System.exit(1);
}
```
- e) writing lines of text
 - print and println methods of PrintWriter class


```
PrintWriter outFile ;
try{
    outFile = new PrintWriter(new FileWriter("file.dat"), true);
    outFile.print("gobber is a gobber");
    outFile.close();
}
catch (IOException e) {
    System.err.println("File Error: " + e.toString() );
    System.exit(1);
}
```

3) tokenizing strings

- a)StringTokenizer tokens = new StringTokenizer(string, char-seperator);
- b)String s = tokens.netxToken();

4) converting to string

- a)int i = Integer.parseInt(string);
- b)fload f = Float.parseFloat(string);

5) FileDialog class

- a)need to implement Frame
- b)getNameBox = new FileDialog(this, "label", FileDialog.LOAD);
- c)getNameBox.show();
- d)string filename = getNameBox.getFile();
- e)TextField nameField; new TextField(30); add(nameField); addWindowListerner(this); nameFiled.setText(filename);

6) System classes

- a)private Buffered kb;
- b)kb = new BufferedReader(new InputStreamReader(System.in), 1);
- c)string line = keyboard.readLine();
- d)string reply = prompt("enter data: ");
- e)[system.out.flush\(\)](#);
- f)[System.err.println](#)
- g)System.exit(#);

7) File class

- a)import java.io.*;
- b)File aFile = new File("path");
- c)to find relative path: String p = aFile.getPath();
- d)to find absolute path: String p = aFile.getAbsolutePath();
- e)file exists? boolean e = aFile.exists();
- f)file is directory? boolean e = aFile.isDirectory();
- g)length of a file? long len = aFile.length();
- h)load filenames in a directory into a string: String [] inDir = aFile.list();

8) Reading from a Remote Site

- a)try {
- b) URL url = new URL(string);
- c) URLConnection link = url.openConnection();
- d) inStream
= new BufferedReader(new InputStreamReader(linkj.getInputStream()));
- e) while((string=inStream.readLine())!=null) {[System.out.print](#)(string);}
- f)}
- g)catch (MalformedURLException e)
- h)catch (IOException e)