Strings and File I/O

Strings

- Java String objects are *immutable*
- Common methods include:
  - boolean equalsIgnoreCase(String str)
  - String toLowerCase()
  - String substring(int offset, int endIndex)
  - String replace(char oldChar, char newChar)
  - int indexOf(String str)

File Input (Text)

- Option 1: Scanner/File
  - Scans input from given File
  - Input from file instead of System.in (keyboard)
  - Easy to scan ints, doubles, etc
- Option 2: BufferedReader/FileReader
  - BufferedReader allows you to read a line at a time instead of a character

File Input (Text)

- Option 1: Scanner

```
Scanner scan = new Scanner(new File(”myfile.txt"));
String s;
while(scan.hasNext()) {//acts like an iterator
   s = scan.nextLine();
}
```

- Option 2: Buffered Reader

```
BufferedReader in = new BufferedReader(new
                                FileReader(”myfile.txt"));
s = in.readLine(); //returns null when EOF reached
while(s != null) {
   s = in.readLine();
}
in.close(); //remember to CLOSE the file!
```

Tips

- Scanner must be imported from `java.util`
- File and Readers must be imported from `java.io`
- Must deal with `IOException`
  - Use try/catch for file operations or declare `throws IOException` in method header
  - `public static void main(String[] args) throws IOException { … }`

Exceptions

- “An exception is an object that defines an unusual or erroneous situation.”
- Examples
  - Divide by 0
  - Array index out of bounds
  - File cannot be found
  - Follow a null reference

try/catch

```
try {
    //statements that may throw an exception
} catch(Exception_Type name) {
    //what to do in case of exception
    //Exception_Type
} catch(Another_Type another_name) {
    //what to do in case of exception Another_Type
}
```

Propagation

```
void divider() throws ArithmeticException {
    int a = 5;
    int b = 0;
    int c = a/b;
}
```

File Output (Text)

- FileWriter allows you to write to a file
- PrintWriter provides interface of System.out
- Remember to import correct packages and handle exceptions

```
PrintWriter out = new PrintWriter(new FileWriter(”myfile.txt"));
out.println(“String 1”);
out.println(“String 2”);
out.close(); //remember to CLOSE the file!
```

Misc…

- Path names
  - Relative path name starts looking in current directory
    - Examples: “myfile.txt”, “mydirectory/myfile.txt”
  - Absolute path name starts from top-level directory
    - Examples “/home/srollins/cs112/myfile.txt”
      “C:\\srollins\\cs112\myfile.txt”
- Binary Files
  - FileInputStream/FileOutputStream read/write *bytes*