

CIST 2372

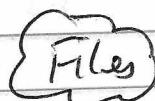
6/18/18

To day	
Files	
Lab 3/A	
Wed.	
more files	
Lab 3/B	
Monday	
Threads	
Lab #4	

comment blocks - at top,
and methods - for
automatic documentation

Exceptions are runtime errors

want to prevent them so code
ends gracefully



Char Sets

+7bit ASCII 8 bits
- one byte 0010,1100 (8bits)
UNICODE - 0001,0101 . 0010,1100 (16bits)

2
4
8
16
32
64
128
 $256 = 2^8$

Java uses UNICODE

text files are ASCII

Files are ASCII
JAVA uses UNICODE

$$65,536 = 2^{16}$$

CIST 2372
6/18/18

import ~~use~~ java.io.*

#1

#2 use try-catch blocks on file usage

* File class *

file f1;
f1 = new File("rout.txt")

f1
size
mod date
permissions
other data about File f1

cannot Read/Write using File

Interrogate file before trying to use -
else will overwrite

long s;
s = f1.length();

boolean b1, b2, b3;
b1 = f1. canRead();
b2 = f1. canWrite();
b3 = f1. exists();

true
false

CIST 2372

6/18/18

import java.io.*

try {

attachfilebf1 File f1 = new File("rom1.txt")

//interrogant file

File Out

Can only

write

full

ascii →

unicode ↙

File OutputStream fout;

fout = new FileOutputStream(f1);

String s1 = "Hello World";

byte barr[] = new byte[11];

barr = s1.getBytes();

fout.write(barr);

fout.close();

set

num

char

os(barr, 0, 11)

}

catch (IOException e) {



AA

BB

CIST 2372

6/18/18

file
cheat sheet



FileInputStream fin;

fin = new FileInputStream(f1)

create byte rarr[] = new byte [st.length()
buf w/rarr]
rarr = fin.read();
|| spaces

String s2 = new String(rarr);

System.out.println(s2);

Class fileDetails {

constructs public fileDetails (String fn);

public static void main (String args) {

~~fd = new~~ FileDetails fd = new FileDetails ("input");