**Chapter 1 Getting Started (Sections 1.1 - 1.4)**

Byte-code

Compiling

Java Virtual Machine

Interpreter

Source Code/Program

Class Loader

.java file

.class file

Syntax errors

Runtime error

Logic errors

Java is case sensitive

Keywords/reserved words

boolean

char

byte/short/int/long float/double

assignment statement =

short hand notation count += 5

assignment compatibilities

Order of operations (+, -, ++, --, ! ) (\*, /, %) (+, -)

Division with integers

++count vs count++

string is an object, not a primitive type

Naming constants: pubic static final double INTEREST\_RATE = 4.5;

Comments: // or /\* \*/

Method descriptions using javadoc: /\*\* \*/ @param @return

Indenting: if/else, loops, methods

System.out.println(); System.out.print();

String Methods:

length( ), equals(other\_string) ,compareTo(other\_string), toLowerCase( ),

toUpperCase( ), charAt(n) (Midterm 1)

equalsIgnoreCase(other\_string), compareToIgnoreCase(other\_string), substring(start), substring(start, end), indexOf(A\_string), indexOf(A\_string, start), trim( )

lastIndexOf(A\_string), (Midterm 2)

lexicographic order (ASCII character set): digit, uppercase letter, lowercase letter

Read Chapter Summary

**Chapter 2 Console Input and Output**

**(Sections 2.1 - 2.3)**

Screen output: System.out.println() and System.out.print()

System,out.println(“The answer is “ + 55);

System.out.printf(“%5d %7.2f %7s”, 45, 77.876, Jose);

45 77.88 Jose

package

import java.text.NumberFormats (be aware they exists and what they are used for)

import java.util.Scanner;

Scanner keyboard = new Scanner(System.in);

value = keyboard.nextInt();

nextByte(), nextInt(), nextLong(), nextDouble(), nextLine(), next()

Always prompt user for input

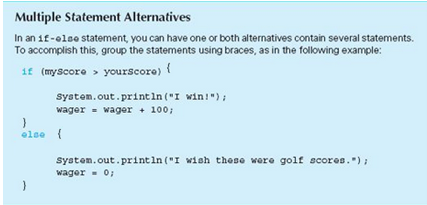
Processing an input file:



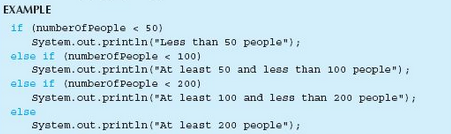
Read Chapter Summary

**Chapter 3 Flow of Control (Sections 3.1 - 3.4)**

if/else statement











Use string1.equals(string2) method to determine if strings are equal, NOT == or !=

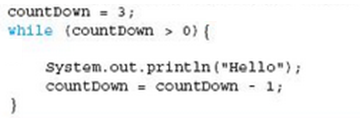
All uppercase letters come before all lowercase letters

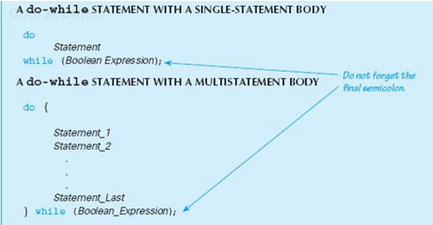
&& AND || OR (short-circuit evaluation)

& AND | OR (complete evaluation)

boolean expressions actually return “true” or “false”

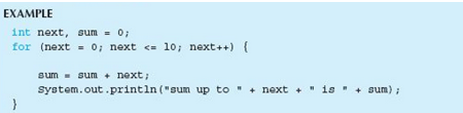
while loop:





Sentinel value – used to end processing





break and continue statement

System.exit(0);

while, do-while, and for loops can be made equivalent

Pitfalls: infinite loops and off-by-one

Tracing variable used for debugging

Assertion checking *assert (n != 0 );*

Write a little bit of code at a time and test it before moving on