0. Read This Before You Begin (XIV)
   a. downloading JDK 7 Update X
   b. installing JDK 7 Update X
   c. modifying PATH and CLASSPATH environment variables

1. Chapter 1 – An introduction to Java and the Java Programming Environment
   a. background on the Java programming language
   b. Java program types:  application, servlet, applet
   c. introduction to OO terminology
      1. objects
      2. encapsulation
      3. instantiation
      4. attributes
      5. behavior
      6. classes  minimal class:  public class MyClass {
      7. methods, method headers  method main:  public static void main(String args[])
   d. keywords: class, public, static, void,
   e. punctuation: ( ) [ ] { } , ; " " ‘
   f. class:  String
   g. variable:  args
   h. Java Development Cycle
      1. Coding
      2. source file
      3. bytecode
      4. class file
      5. translating
   i. compiling
   j. interpreting
   k. intermediate language (virtual machine language)
   l. machine language
   m. syntax errors, logic errors, run-time errors
   n. platform independence  ("write once, run anywhere...")
   o. Using the command line to compile/run programs
      1. javac javac MyClass.java  //calls the Java compiler
      2. java java MyClass  //calls the JVM (interpreter) to run the program
      3. drive letter, path  c:\cis_103\assignments
      4. Java Virtual Machine (JVM)  This is actually the JRE (Java run-time environment) – an interpreter

2. Chapter 2 – Variables, Constants, Operators, and Writing Programs Using Sequential Statements
   a. variables  [discuss notion of identifiers]
   b. identifier requirements in Java  ($ _ letters digits), no spaces, can’t begin with a digit, CASE SENSITIVE
   c. Java data types  [discuss value and reference types]
      1. value (PDT’s)  //stores an actual PDT value. Java has 8 primitive data types
         byte, short, int, long, float, double, char, boolean
      2. reference  //stores a reference (address) of a reference type (object)
      3. integer/real  //byte, short, int, long (integral) / float, double (floating-point)
      4. Strings  //reference type). A String is an object in Java
   d. declaring and initializing variables
   e. arithmetic and assignment operators
      • precedence and associativity
   f. the assignment statement
   g. input in Java using the Scanner class and methods
   h. output in Java using the System.out object and methods print, printf, and println
   i. interactive  input statements (javax.swing.JOptionPane.showInputDialog)
   j. interactive  output statements (javax.swing.JOptionPane.showMessageDialog)
      • System.exit(0); statement is used in GUI programs. Otherwise it is not needed!
3. Chapter 3 – Writing Structured Java Programs
   a. flowcharts and pseudocode
   b. using Java Wrapper classes
      1. Integer.parseInt( <String> )
      2. Double.parseDouble( <String> )
   c. writing sentinel-controlled loops
   d. Java comments // and /* */
   e. local and global variables
   f. named constants
   g. calling a method [modularization]
   h. static keyword

4. Chapter 4 – Writing Programs That Make Decisions
   a. boolean operators
      1. relational operators
      2. logical operators
   b. relational and logical operator precedence and associativity
   c. comparing Strings
      1. Strings are objects (reference type)
      2. using equals and compareTo methods
      3. == versus equals()
         1. comparing references
         2. comparing the contents of objects (Strings)
   d. decision statements
      1. if
      2. if-else
      3. nested if statements
   e. flow of control
   f. block [compound statement]
   g. logic errors
   h. null statement [ ; ]
   i. switch statement [ case structure ]
   j. using decision statement to make multiple comparisons
      • and/or logic

5. Chapter 5 – Writing Programs Using Loops
   a. increment and decrement operators ( ++, -- )
      1. prefix / postfix form
   b. writing a while loop in Java
      1. loop control variable
      2. boolean comparison
      3. sentinel value
      4. infinite loop
      5. null statement
   c. using a counter to control a loop
   d. using a sentinel value to control a loop
   e. writing a for loop in Java
   f. writing a do..while loop in Java
   g. nesting loops
   h. accumulating totals in a loop
      • discuss variables used as counters and accumulators
   i. using a loop to validate input
Chapter 6 – Using Arrays in Java Programs
a. basic array concepts
   1. declaring
   2. initializing
   3. accessing
b. bounds-checking
   1. exceptions ArrayIndexOutOfBoundsException
c. using named constants with arrays
d. searching an array for an exact match
e. parallel arrays

Using Methods in Java Programs
a. simple Java method
b. parameters
c. arguments
d. method header
e. parameterized methods (single and multiple parameters)
f. writing methods that return a value
g. passing an array element to a method
h. passing an array to a method
i. pass by value and reference
j. overloading methods
   1. method signature
   2. polymorphic code
k. using Java's built-in methods [Java API]
   1. Math and String class methods
   2. printf method (and coverage of format specifiers/flags)

Writing Control Break Programs
a. single level
b. accumulating totals
c. multiple levels
d. [control break fields – notion of a "previous" variable]