pre-post conditions:

<https://www.cs.umd.edu/class/fall2002/cmsc214/Projects/P1/proj1.contract.html>

Lesson Plan:

1. Week 1 (Bk 1, pp 1-19)
   1. Basic terms
      1. access modifiers
   2. How It Works
   3. Basics of OOP
   4. Pseudocode
   5. Comments
   6. Variables (and reference variables) and data types (numeric and non-numeric, non-primitive vs primitive)
   7. Procedural Abstraction
   8. Operators, escape sequences, String operators, shortcut operators
   9. Breaking Down the Code
   10. Division and rounding
   11. Division by zero
   12. Explicit/Implicit type casting
   13. Different kinds of errors (syntax (compile), runtime, logic)
   14. order of operations
2. Week 2 (Bk 1, pp 20-29)
   1. Class Basics and benefits, class memebers
   2. Obj references
   3. Instantiating the object
   4. Instance variables
   5. Methods + calling methods
   6. Encapsulation!!!
   7. API basic explain
   8. Creating objects
   9. Constructors,
   10. Using obj references
3. Week 3 (Bk 1: pp 29-54)
   1. java Class Library
   2. packages and import
   3. String
   4. Random
   5. Scanner
   6. System/PrintStream
   7. DecimalFormat
   8. Math
   9. Object wrapper
   10. JOptionPane
   11. CSD in jGRASP
4. Week 4 (Bk 1: 55-63)
   1. flow of control
   2. selection
   3. equality, relational operators
   4. logical operators
   5. more operator precedence
   6. DeMorgan’s
   7. simple if
   8. if/else
   9. if/else if
   10. sequential if/else
   11. nested if/else
5. Week 5 (Bk 1, 64-70)
   1. dangling else
   2. testing: test plan
   3. pre/post conditions
   4. assertions
   5. developing appropriate test cases including boundary cases
   6. unit testing
   7. integration testing
   8. Comparing Floating-point numbers
   9. the equals method
   10. compareTo
6. Week 6 ( Bk 1, 71- bk 2 p 1)
   1. switch/case
   2. While loops + sentinel
   3. priming/pumping
   4. using File class
   5. Scanner class
7. Week 7 (Bk 2 pp 2-13)
   1. Accumulation
   2. counting items
   3. calculating an average
   4. finding max/min
   5. type-safe input using scanner
8. Week 8 (bk 2 pp 13-19)
   1. counstruction loop conditions
   2. testing while loops
   3. do/while
9. Week 9 (bk 2 pp 20-31)
   1. For loops
   2. local variables
   3. nested loops
   4. flag vars
   5. how to write a method
   6. using input args[]
   7. helper methods
   8. formal vs actual parameters
   9. Project 3
10. Week 10 (bk 2 pp 31-41)
    1. User-defined classes
    2. defining a class
    3. fields of class
    4. client programs
    5. instance vars
    6. writing class methods
    7. writing constructors
    8. method overloading !!!
    9. signature of a method
    10. writing accessor methods
11. Week 11 (bk 2 pp 41-44)
    1. writing mutator methods
    2. Encapsulation!!!
    3. writing data manipulation methods
12. Week 12 (bk 2 pp 45-end)
    1. Obj ref ‘this’
    2. implicit para
    3. method overriding!!
    4. static meaning
    5. instance method
    6. UML diagrams
13. Week 13 (Bk 3 pp 1-9)
    1. 1-d arrays
    2. declaring and instantiating
    3. assign and access elements
    4. printing and reading elem
14. Week 14 (bk 3 pp 9-23)
    1. summing array elem
    2. min/max of array
    3. copying arrays
    4. changing array size
    5. comparing arrays
    6. using arrays in user-defined cases
    7. clients + access to array elem
    8. Retrieving command line args
15. Week 15 (bk 3 pp 23-39)
    1. Sequential search of unsorted array
    2. selection sort
    3. insertion sort
    4. sorting an array of objects
16. Week 16 (bk 3 p 39)
    1. Midterm
    2. Flex
17. Week 17 (bk 3 pp 40-43)
    1. sequential serach of sorted array
    2. binary search
    3. call by value
    4. Project 5
18. Week 18 (bk 3 pp 44- bk 4 p1)
    1. Hierarchies
    2. super/subclasses
    3. inheritance
    4. direct super/subclass
    5. overriding methods
    6. inherited class members
    7. protected access modifiers
    8. superclass constructors
       1. implicit invocation
    9. subclass constructors
19. Week 19 (bk 4 pp 2-13)
    1. Adding specialization to the subclass
    2. overriding inherited methods
    3. the protected access modifier (more on that??)
    4. abstract classes and methods
       1. restrictions
    5. polymorphism
       1. dynamic or late binding
       2. downcasting
    6. Project 6
20. Week 20 (bk 4 pp 13-14)
    1. Selection sort
    2. Bubble sort
    3. mergesort
21. Week 21 (bk 4 pp 15-17)
    1. Multidimensional arrays
    2. intro to walls and mirrors
    3. into to recursion and functional decomposition
22. Week 22 (bk 4 pp 17-24)
    1. Recursive helper methods
    2. recurrence relations
    3. box trace
    4. factorial
    5. string backwards
23. Week 23 (bk 4 pp 39 – bk 5 p1)
    1. Recursion and efficiency
    2. ADT
       1. modularity
       2. procedural abstraction
       3. specifications
       4. implementation
       5. data structure
    3. information hiding !!!
    4. the ADT list
       1. operations
    5. choosing data structures
24. Week 24 (bk 5 pp 2-5)
    1. Interfaces
    2. interface vs abstract classes
    3. basic exceptions
    4. catching exceptions
    5. checked vs unchecked
    6. the following:
       1. ArithmeticException
       2. NullPointerException
       3. ClassCastException
       4. ArrayIndexOOBException
       5. IllegalArgumentException
       6. IndexOOBException
    7. Information hiding vs encapsulation
25. Week 25
    1. java.util.List<E>
       1. methods in interface
    2. basic insertion/traversal/deletion
    3. basics of generics
    4. what is/how to use ArrayList class
26. Week 26
    1. Decimals
    2. Base 2: Binary numbers
    3. hexadecimals
    4. octals
27. Week 27: Elements of Design
    1. Bottom-up
    2. Top-down
    3. modularity
    4. reusable code
28. Week 28: Computing in Context
    1. System Reliability
    2. Privacy
    3. Legal Issues + intellectual property
    4. social + ethical ramifications of computer use
29. Week 29: Flex/Review
30. Week 30: Final