I. Object-Oriented Design

- Classes
  - Classes vs. Instances
  - Fields & Methods
  - Cohesion and Coupling
  - Class vs. Instance scope of fields and methods
  - Classes with a fixed number of instances: enums

    Example: the Checkers application

- Inheritance
  - Inheritance hierarchies
  - Inheritance of instance fields
  - Inheritance of instance methods, overriding
  - Preventing inheritance of classes, methods: final

    Example: the Company application

- Access control

- Encapsulation
  - Instance fields
  - Accessor & Mutator methods
  - Inner classes
  - Documenting the public API

- Abstract classes
- The diamond problem
- Interfaces

  Example: the Superheroes application

- Scoping and binding in Java:
  - Scope of a variable, shadowing
  - Methods and signatures
  - Overloading methods
  - Polymorphism

- Important classes and interfaces
  - the Object class: equals, hashCode, toString
  - the Comparable interface: equals, compareTo, hashCode
  - the Serializable interface: serialVersionID

II. Input/Output and Exception Handing

- Preconditions

- Exceptions
  - Reporting
  - Handing
  - Try – catch – finally

- The exceptions hierarchy in Java
  - Checked vs. unchecked exceptions
  - Defining your own exceptions

    Example: the Roster application

- Files and Streams

  - Fundamental vs. object data types: when is the size of the object known?
  - ASCII (character) vs. binary (byte) files
  - End of file – integer -1

  - FileReader / FileWriter – ASCII files
  - FileInputStream / FileOutputStream – binary files

  - BufferedReader/BufferedWriter – optimized performance
  - System.in, System.out, Systerm.err – streams
  - Sequential vs. Random Access (RandomAccessFile)

  - Serialization and De-serialization: classes must implement Serializable

  Example: the StreamsDemo

III. Data Structures and Algorithms

- Fundamental data structures
  - Array
  - ArrayList
  - Linked list
  - Stack
  - Queue

  Examples: LinkedList, StackAndQueueDemo

- Recursion

- Sorting
  - Record = key + additional description information
  - Total ordering on key
  - Stable sorting: no re-ordering of equal elements
  - Sorting algorithms
    - countingSort
    - selectionSort
    - bubbleSort
    - mergeSort

  Example: Sorter

- Searching
  - Sequential search, Binary search

- Binary trees
  - Structure of a binary tree
  - Binary search trees
  - Tree traversal: preorder, inorder, postorder
  - Search, delete arbitrary elt, insert, successor, predecessor, min, max

- Heaps
  - The heap property
  - Insert, delete min (root), heapify, heapSort

  Example: MinHeap

- Sets and Maps

  Example: SetAndMapDemo

IV. Additional Topics

- Direct addressing vs. hashing
  - Computing hashCodes
  - Chaining
  - Collisions

- Passing arguments to a method
  - fundamental vs. object data types

- Relational databases
  - The differences between the relational and OO model
  - Keys: primary keys, superkeys, primary keys
  - Referential integrity: foreign keys

- SQL queries
  - Simple select – project - join queries, with order-by
  - Using the distinct keyword

    Select [distinct] <attribute list>
    From <table list>
    Where <conditions>
    Order by <column list or positional indexing>

  Example: University.sql