Overview.

Remember this list is not meant to be comprehensive BUT is done to help you study.  (it doesn't mean that other lecture slide/exercise topics won't be in the final!)

Loops
- Write for loops/for-each loops/while loops
- Write code that iterates over an array to do something with the elements in the array (e.g., find patterns, count elements, etc).
- Write nested loops to work with 2D arrays (again, find patterns, calculate properties of the data, etc)
- Reason about the behavior of nested loops by tracing the code.
- Write code that iterates over ArrayLists and ArrayList<T> (again to do something interesting with the data stored within).
- Identify errors in loop conditions
- Write boolean expressions (using &&, ||, <, >, ==, etc) to correctly control loops

Strings
- == vs .equals() (do not use .match() it doesn’t do what you think it does)
- Use common String methods (substring(), charAt(), toCharArray(), etc )
- Explain how strings are immutable. Trace code that “modifies” strings by constructing new strings. Write code to “modify” a string by constructing a new string.

File IO
- Scanner class
- has* methods vs next* methods

Objects, Classes, Inner Classes and Inheritance
- Describe the difference classes vs objects
- Draw memory modes with objects (including scope)
- Use and draw memory models for inner classes
- Describe the difference between memory models and subclasses
- Describe the difference between overloaded and overridden methods
- Know how to write a subclass
- Know what Access Level Modifiers (ie. private, public, protected) do
- Determine if a particular method can access given members in other classes based on the given members Access Level Modifier (ie. private, public, protected)
- Describe the difference between Static and Non-Static members and methods
- Explain at a high level what polymorphism is
- Determine which overridden method is called at runtime
- Describe what the toString() method does

Casting
- Know what casts do
- Casting primitives (ie int, double, etc) vs casting references to objects
- Know when a cast (or lack thereof) results in a compile time error
• Know when a cast results in a runtime error
• Use casting correctly

Abstract classes and Interface
• Describe when a class must be abstract
• Write an abstract method
• Know how to define an interface
• Know how to use an interface
• Know the difference between an interface and an abstract class
• Know how to extend and abstract class or implement an interface

GUI/Applet stuff
• Know what is a MouseListener, ActionListener, KeyListener
• Know how to define a custom listener by extending Mouse/Action/KeyListener
• Know how to add a listener to a container (i.e., a JPanel) using the appropriate method

Generics
• Know what the T in ArrayList<T> means
• Know what can be assigned to generic references

Recursion
• Write simple recursive methods
• Trace recursive methods, including drawing stack frames
• Identify the base case(s) (Yes, there can be more than one!)
• Identify the recursive call(s) (Yes, there can be more than one!)
• Know the basics of recursion
  a. need a base case
  b. need to recurse on a smaller problem (eventually working toward the base case)

Threads
• Know the basic concept of what a thread is
• Know the basic thread methods (start(), sleep(), etc)
• How to create your own thread.
• Describe what happens when you put the main GUI thread to sleep

Exceptions
• Know what an exception is
• Writing a try/catch block
• Difference between Checked and Unchecked Exceptions