CSE1322L Final Exam – List of Topics

To perform well on CSE1322L Final Exam, students should be thoroughly familiar with the topics listed below. For in-depth knowledge of the topics, students should refer to the course textbooks, lecture slides, D2L videos/example code, in-class lectures/recitations, and quizzes.

Students who are uncomfortable with any of the topics should reach out to instructors, GTAs and tutors.

For practice problems and examples see the old exams on our website.

Given the following topics, students should be prepared to define terms, trace code (Java/C#) and provide output, and code simple programs:

- **Module 1:**
    - Variables (primitive vs complex)
    - Reading/Writing to the console
    - Conditionals (if/then/else if/else, case)
    - Logical operators (<, <=, >, >=, ==, !=, and, or, not)
    - Arithmetic operators (+, -, *, /, %, +=, -=, *=, /=, ++, --, difference between x++ and ++x).
    - Methods (declaring, return types, parameters, calling, arguments)
    - Loops (for, while, do, foreach)
    - Operator Precedence (i.e. Order of Precedence)
    - Arrays (defining, accessing cells, traversing, array properties such as length)
    - Two dimensional arrays (defining, accessing cells, traversing, properties)
    - Classes (defining, attributes, getters/setters, writing methods, instantiating (new), encapsulation, constructors (default & overloaded), static keyword (methods & attributes))
    - Driver (“dot” . operator (for methods and attributes), calling methods, accessing attributes in objects/classes).

- **Module 2:**
    - ArrayLists or Lists (Defining, adding, accessing, removing items, getting size, iterating through)
    - Passing by Value (especially tracing code with arguments going to methods)
    - Passing by Reference or passing an object.
    - Returning values from a method
    - Garbage, how it’s made and dealt with
    - Method Overloading

- **Module 3:**
  - Inheritance (what can be inherited, why we use inheritance, instanceof, is keywords)
  - Overriding methods (e.g. toString()/ToString)
  - Object/object
  - this vs super/base
  - Public/Private/Protected
  - Polymorphism (What it is, and how to use it, Relationship to inheritance, Late Binding)
  - Abstract classes (How/when to use them, understanding mix of abstract and concrete methods,
  - Interfaces (How/when to use them, understanding they have only abstract methods)
  - Casting (How it's used)

- **Module 4 (GUIs and Graphics):**
  - What is a GUI
  - JavaFX and Windows Forms
  - Differences between Backend, Middleware and Frontend
  - Different types of GUI components
    - Layouts
      - Grids, Tables, Tabs, Scroll and Cards.
    - Controls
      - Text Areas, Buttons, Selection Lists, Radio Buttons, Checkboxes, Knobs, Sliders, Links, Date Picker, File Chooser, Links.
    - Labels, Progress Bars, Canvas
  - GUI coordinate systems
  - Colors
  - Drawing
    - Java: strokeLine, fillOval etc
    - C#: DrawLine, fillEllipse etc
  - Event Handlers
    - What is an event
    - What types of events exist
    - What a handler is

- **Module 5 (Recursion):**
  - What is the function stack
  - How to trace code
  - How variables are local to each method call by default
  - Recursion
    - Base Condition
    - Recursive Calls
    - Tracing Recursion
    - Infinite Recursion and how to avoid it
    - Solving problems recursively.

- **Module 6 (Exception Handling):**
    - What is an exception
    - Catching an exception
        - try
        - catch
        - finally
    - Exception object hierarchy
    - Throwing exceptions
    - Defining custom exception
    - Checked vs Unchecked exceptions.

- **Module 7 (File IO):**
    - How to open a file for reading, writing, appending, creating
    - How to read a text file, and write to a text file.
    - Parsing file of different formats
    - Reading a binary file
    - Closing files
    - Reading from and Writing to a socket

- **Module 8 (Threads and Parallel Processing):**
    - Dividing work into multiple threads
    - Understanding why 3 threads doesn’t mean completing a task in ⅓ the time.
    - Using Runnable/ThreadStart, starting a Thread
    - Thread States
    - Task Parallel Library/Parallel.ForEach, Parallel.Invoke
    - ForkJoin Pools, Recursive Actions, Recursive Task

- **Module 9 (Stacks, Queues, Linked Lists):**
    - Creating Linked Lists (including adding, deleting and searching nodes)
    - Creating Doubly Linked Lists (Adding at the front, middle or back, removing from the front middle, or back)
    - Working with a Stack, push, pop and peek
    - Working with a queue, enqueue, dequeue, peek
    - Understanding trees and graphs at a high level.