SA - short answer | FR - free response | CT - code tracing | CW - code writing

UNIT 1

Early Definitions (SA/FR)

- Algorithms and Abstraction
- Basic data values/types
- Syntax, runtime, and logical errors

Computer organization (SA/FR)

- Reading and/or reorganizing bytecode
- Binary addition and/or conversion
- Data abstraction - pixels or ascii
- Data abstraction - design your own abstraction
- Comprehension of half-adder/full-adder/n-bit adder/flip flop
- How files work at a basic level

Basic Programming (SA/FR)

- Print vs Return
- Basic scope, without crazy global stuff
- Evaluate logical operations
- Evaluate range expressions
- Comprehension of linear search
- Indexing/slicing with strings

Basic Programming (CT/CW)

- Code tracing basic algorithms
- Code tracing with variables
- Code writing functions with variables and basic data values
- Code writing - convert control flow chart to code
- Code trace code with control structures (conditionals and loops)
- Write new code with control structures (conditionals and loops)
- Convert between boolean expressions, truth tables, and circuits
- Implement linear search algorithms
- Code tracing - indexing/slicing/methods with strings
- Basic code writing with strings
- Using basic string methods - isalpha(), isdigit(), lower(), upper()
- Input() and file reading/writing

UNIT 2

Graphics (CW)

Lists (CW)

- Create lists
- Add to lists (append, insert, +)
- Remove from lists
- Check if an element is in a list
- Swap or replace elements
- Iteration over elements of lists
- Using 2D lists (CT)

Destructive vs non-destructive methods (CW)

- Examples of each in python (append, etc)
- Identifying destructive methods
- Implementing a given algorithm destructively or non-destructively

Aliasing and Mutability (CT)

Recursion

- Parts of recursive functions (SA)
- Tracing recursive functions (CT)
- Writing very simple recursive functions on lists, strings, ints (CW)

Runtime and Big-O Notation

- Differences between runtimes (SA)
- Runtime Analysis of simple Loops (CT)

Search Algorithms

- Recognizing linear/binary search (CT)
- Recalling runtime of different search algorithms (SA)

Sorting Algorithms (Insertion sort, Selection sort, Mergesort)

- Runtime of given sorting algorithms (SA)
- List sorting (CT)

Hash Tables/hash functions (FR/SA)

Dictionaries (CW)

- Purpose and use cases
- Creation
- Adding elements
- Removing
- Replacing values
- Iteration over

Trees

- Vocabulary (parent, children, leaf, root) (SA)
- Implemented as dictionaries (CW)

  - Trees vs Binary trees vs BSTs (comparison & runtime) (SA)
  - Very simple tree operations with given algorithm (CW)
- Graphs (CT)
  - Implemented as dictionaries or adjacency matrices
  - Breadth First Search and Depth First Search
- Tractability (SA)
  - Runtime analysis
  - Complexity classes P and NP
  - Implications of P=NP

UNIT 3

- Hardware scaling (SA/FR)
  - Circuit-level Concurrency
  - Multiprocessing
  - Pipelining
  - Multitasking
  - Distributed Computing
  - MapReduce
  - Challenges of implementation
- The Internet (SA/FR)
  - Packets
  - Routers
  - IP Addresses
  - DNS
- Security and Privacy (SA)
  - Adversaries
  - DDOS attack
  - Man in the middle attack
  - Malware
  - Authentication - Certificates, Usernames/passwords
  - RSA
- Encryption (CT)
  - Caesar
  - Substitution
- The Cloud (SA)
  - Software as a Service
  - Platform as a Service
  - Infrastructure as a Service

UNIT 4

Data Analysis (CW)

- Opening and reading a file
- Parsing strings
- Converting strings to ints to floats and vice versa
- Dataframe - how it organizes data, not how to call it (SA)

Simulation (CT)

- Identifying components and rules of a model (SA)
- Representing a model as a dictionary
- Visualizing a model using tkinter
- Updating a model with a time loop
- Events - key presses and mouse clicks

Data Summarization (CW)

- Algorithms to find means, medians, counts

Machine Learning (SA)

- Define ML and AI
- Define training vs. testing data
- When to use Classification vs regression
- Identify basic classification algorithms and when they can be used

Visualization (SA)

- Identify category vs. ordinal vs. numerical data
- Correct use of plots
- Which plots should be used for
  - counts (histograms),
  - category (x-axis, independent variable) and numerical data (y-axis, dependent variable) (bar charts),
  - numerical and numerical data (scatterplots)

Monte Carlo

- Define true randomness vs. pseudo-randomness (SA)
- Write very basic Monte Carlo code (CW)
- Recognize the relationship between # trials, accuracy, and time (SA)

AI (SA)

- Types of AI agents
- Graphs to model states and actions
- BFS to search graphs

UNIT 5

CS History (SA)

- Identify math theories that influenced CS (binary numbers & boolean algebra)
- Identify hardware components that made personal computing possible (transistor, integrated circuit, and microprocessor)
- Recognize the contributions of a few of the founders of CS:
    - Charles Babbage [first general computer design]
    - Ada Lovelace [first program & computational thinking]
    - Claude Shannon [circuit design & information theory]
    - Alan Turing [Turing machines & halting problem]
    - John Mauchly & J. Presper Eckert [ENIAC, the first computer]
    - John von Neumann [software architecture]
    - Grace Hopper [the compiler]
    - Douglas Engelbart [the mouse & the GUI]
    - Vinton Cerf & Robert Kahn [TCP/IP]
    - Tim Berners-Lee [HTML & URL]

CS Ethics (SA)

- Identify our three big rules for code maintenance
- Define the data economy and explain why it exists
- List recent examples of bias in machine learning
- Recognize where there can be room for debate about ethics in AI

CS Future (SA)

- Define what a deepfake is
- Recognize how 5G will be different from 4G and predecessors
- Identify occupations that may be at risk because of automation
- Define quantum computing
- Define the Turing Test and its purpose