CS50 Week 0 - Introduction to Computer Science and Programming

Animesh Garg and ChatGPT

1 Course Overview

- **CS50** is Harvard's introduction to computer science and programming.

- Emphasizes **problem-solving** rather than just programming languages.

- Designed for students with **no prior experience** in CS.

- Grading is based on **personal progress**, not comparison with classmates.

- The course includes:

  - **CS50 Puzzle Day** (team-based problem-solving event).
  - **CS50 Hackathon** (overnight final project sprint).
  - **CS50 Fair** (end-of-course project showcase).

2 What is Computer Science?

- **The study of information**: How it is represented and processed.

- **Computational thinking**: Applying CS concepts to real-world problems.

- **Problem-solving process**:
$$\text{Input} \rightarrow \text{Processing} \rightarrow \text{Output}$$

3 Binary and Data Representation

3.1 Counting in Different Number Systems

- **Decimal (Base 10)**: Uses digits 0-9.

- **Binary (Base 2)**: Uses only 0 and 1.

  - Each binary digit (bit) represents **on (1) or off (0)**.
  - A byte consists of **8 bits** and can store values **0-255**.
  - More bits = ability to store larger numbers.

3.2 Text Representation

- **ASCII (7-bit, later 8-bit encoding)**: Maps numbers to letters.

$$\text{Example: 'A'} = 65, \text{'B'} = 66$$

- **Unicode (modern standard)**:
  - Backward-compatible with ASCII.
  - Supports **all human languages & emoji**.
  - Uses **16-bit, 24-bit, or 32-bit encoding**.

3.3 Color Representation

- **RGB model**: Each color is a mix of Red, Green, and Blue.

- **Each pixel** on a screen is stored as 3 bytes (24 bits total).

$$(72, 73, 33) \rightarrow \text{Yellowish color}$$

3.4 Other Data Types

- **Images**: A collection of colored pixels.

- **Videos**: Sequences of images displayed at high speed.

- **Sound**: Represented using frequency, duration, and volume.

4 Algorithms

- **Definition**: Step-by-step instructions to solve a problem.

- **Efficiency Matters**:
  - **Linear Search** (page-by-page search) $\rightarrow$ **Slow** $O(n)$.
  - **Binary Search** (divide and conquer) $\rightarrow$ **Fast** $O(\log n)$.

4.1 Example: Phone Book Search Algorithm

1. Open the book to the **middle page**.

2. If the person is on the page $\rightarrow$ **Done**.

3. If the person is earlier, search the **left half**.

4. If the person is later, search the **right half**.

5. Repeat until the person is found or the book is exhausted.

5 Programming Concepts

5.1 Key Building Blocks

- **Functions**: Perform specific actions (e.g., `say "hello"`).

- **Loops**: Repeat an action (`repeat 3 times` or `forever`).

- **Conditionals**: Make decisions (`if X then Y else Z`).

- **Boolean Expressions**: True/False conditions (`if touching wall`).

5.2 CS50's AI & Debugging

- **Rubber Duck Debugging**: Talking through a problem often helps find errors.

- **CS50 AI**: A **course-specific chatbot** to assist students.

6 Scratch (Block-Based Programming)

- **Why Scratch?**

  - No syntax errors (no parentheses, semicolons, etc.).
  - Focuses on **logic and structure**.
  - Prepares students for traditional programming languages.

6.1 Basic Scratch Elements

- **Sprites**: Characters or objects in the program.

- **Scripts**: Code that controls sprites.

- **Events**: Actions that trigger scripts (`when green flag clicked`).

- **Motion**: Moves sprites on x-y coordinates (`move 10 steps`).

- **Loops**: Repeats actions (`forever` or `repeat 10`).

- **Conditionals**: Controls logic (`if touching sprite then do X`).

6.2 Example: Making a Cat Meow

1. Use `when green flag clicked` event.

2. Add `play sound "meow"` action.

3. Wrap it in a `repeat` loop for multiple meows.

7   Game Development in Scratch

- **Example: "Oscartime"**

  - Drag falling trash into Oscar's trashcan.
  - Uses **loops, conditionals, random positioning, and scoring**.

- **Example: "Ivy's Hardest Game"**

  - Player moves around avoiding obstacles.
  - Uses **keyboard input and AI-like behavior** for enemy movement.

8   Final Takeaways

- **CS50 is about problem-solving**, not just programming.

- **Binary enables everything**: Numbers, text, colors, sound, and images.

- **Algorithms matter**: **Efficiency is key** to handling large problems.

- **Scratch simplifies coding concepts** before transitioning to C and Python.

- **Next Steps**: Start **Problem Set 0** by creating your own Scratch project.