Topics we covered:

- **The basics:**
  - Simple data types: `int`, `float`, `double`, `char`
  - Operators, expressions, statements, computer arithmetic
  - Basic I/O - `cin` and `cout`
  - Simple program structure, the preprocessor

- **Decision statements:**
  - Logical operators and expressions
  - `if` statement
  - `if-else` statement
  - Nested `if` statements
  - `switch-case` statement

- **Looping statements:**
  - Pre-test statements: `while`, `for`
  - Post-test statement: `do-while`
  - Looping examples and techniques: summation, iteration

- **Functions:**
  - Function syntax
  - Arguments, call-by-value, call-by-reference
  - Return values
  - Variable scope - local vs global

- **Arrays:**
  - Array declaration and syntax
  - Array initialization, array input/output
  - Array techniques, arrays and (for) loops
  - Arrays as function arguments
  - C-style strings (arrays of `char` terminated with the `NULL` character)
  - 2D arrays: rowwise vs columnwise operations

- **Files (streams):**
  - The standard streams: standard input `cin`, standard output `cout`, standard error `cerr`
  - Input streams (`ifstream`es)
  - Output streams (`ofstream`es)
• structs
  – struct declaration - a new type
  – Accessing members of structs
  – structs and arrays, arrays of structs

• Classes:
  – Concept of an “object”
  – Class members and methods - syntax
  – public and private class components
  – Class declaration, header files
  – Pointers to classes, dynamic class instantiation
  – Class examples, the C++ string class

• Pointers:
  – Pointer variables, declaration
  – The pointer operators, & and *
  – Arrays and pointers
  – Pointers and function arguments, call-by-reference
  – Dynamic memory, the new and delete operators

• Data Types - internal representation:
  – Base conversion: 2, 8, 10, and 16
  – Binary arithmetic

• Linked lists:
  – Dynamic memory allocation of linked list nodes
  – the -> operator
  – declaring and initializing linked lists
  – Linked list operations: inserting and removing nodes, traversal
  – Deleting nodes

• Recursion:
  – Basic concepts - a function that calls itself
  – Necessary conditions:
    1. Problem can be expressed as a smaller version of itself
    2. A stopping case
  – How implemented: activation records, a stack

•Searching and Sorting:
  – Searching - linear, binary
  – Sorting - Basic ideas