January 27, 2025

LEC 08 - Abstract Data Types

Abstraction

Key Idea: View objects as entities that can store data and operations (useful to solve problems)

- Focus on semantics
  - Hide details from user
  - Freedom to design or update algorithms
  - Independant of programming language

Abstract Data Types

- In Computer Science, we recycle our intuition about the outside world ADTs
- We abstract data and operations, and suppress the implementation

Examples:

- Sequences of items
  - Can be added, removed, accessed by position, etc.
- Specialized collection of items where we only have access to the most recently added item
- Collection of items accessed by the associated keys

Stacks

- In Python, frames for function calls form a stack

What does a Stack store?

- Any object, same as a standard list

What are the operations?

- `push()` → Add an item to the top of the stack
- `pop()` → Remove the top-most item from the stack
- `is_empty()` → Return `True` if the stack is empty

Stack Application - Balanced Parantheses

- In some situations, it is important that opening and closing brackets match
  - IDE checking for well-formed code
  - Compilers, interpreters, calculators, etc.

January 27, 2025

LEC 08 - Abstract Data Types

Define Balanced Parantheses:

- A string with no parentheses is balanced
- A string that begins with a left parentheses and ends with a right parentheses and has balanced parentheses inbetween is balanced
- The concatenation of two strings with balanced parentheses is balanced
    - (...)(...)

Check for “Balancedness”:

- Stacks can be used to check balanced parentheses