CSE 373

MARCH 27 – COURSE INTRODUCTIONS;
ADTS; STACKS AND QUEUES
WELCOME!

• Administrative Minutiae
• Course Objectives
• Review of Stacks and Queues
• Abstract Data Types (ADT)
WAITLIST/OVERLOAD

- Please fill out the survey (will also post link on the course website)
  - [https://catalyst.uw.edu/webq/survey/cseadv/328147](https://catalyst.uw.edu/webq/survey/cseadv/328147)
- Fill this out as quickly as possible
COURSE INFO

• Evan McCarty (ejmcc@uw.edu)

• Office hours (CSE 214)
  • Mondays: 11:00 – 11:50
  • Wednesdays: 3:30 – 4:20
  • By appointment or over email
COURSE STAFF

• TAs are posted on the course website, along with their office hours
• TAs for sections will be posted by Wednesday
• Office hours will start this Friday
HOMEWORK

- Homework will be assigned on Wednesdays after class and due the following Wednesday at midnight via canvas
- HW 1 out this Wednesday (3/29)!
- Late Policy :
  - 15% per day late
  - Max 3 days
HOMEWORK

- Implementation is only part of the problem
- Write-up/written questions also important
- Separate submissions on canvas (more on Wednesday)
- Regrade requests
HOMEWORK

• Academic honesty
  • High level discussion
  • Fully understand submission

• Reasonable effort and office hours
LECTURES

- Lecture slides will be posted online after class
- Questions are strongly encouraged
- All material fair game for exams
- Weiss textbook
SECTIONS

• Conducted by TAs
• Practice problems
  • Analysis and Implementation
• Supplementary instruction
EXAMS

• Midterm exam (20%)
  • 2:30 – 3:20; Friday, April 28
• Final Exam (30%)
  • 2:30-4:20; Tuesday, June 6

• Exam review in lecture before exams
PRE-HW1 TO DO

• Set up the JDK
• Install Eclipse (not required, but recommended)
DATA STRUCTURES AND ALGORITHMS

• Understand and recognize behavior of key data structures
• Understand and solve common data structure problems
• Analyze operations and algorithms
• Implement data structures and understand design trade-offs
CSE 143

- Object-oriented Programming
  - Classes and Inheritance
  - Methods, variables and conditions
  - Loops and recursion
  - Linked lists and simple trees
  - Basic Sorting and Searching
  - Concepts of Analysis $O(n) \text{ vs } O(n^2)$
  - Client v. Implementer
CSE 373

• Design decisions
• Critical thinking
• Implementations
• Debugging and Testing
• Abstract Data Types
ABSTRACTION

• Software engineering v. Computer Science

• Applicable across languages and implementations

• Behavior focus
  • How can you recognize an ADT?
DEFINITIONS

• Abstract Data Type (ADT)
  • Operations and expected behavior
• Data Structure
  • Specific organization of data
  • Can be analyzed
• Implementation
  • Language specific application
DESIGN DECISIONS

• Between an ADT and its implementation, there are design decisions

• Constraints of the problem
  • Memory v. Speed
  • One function v. another
  • Generality v. Specificity
DESIGN DECISIONS

- Linked List v Array
  - Overhead
  - Memory use
  - Adding to middle
  - Traversal
  - Insertion
DESIGN DECISIONS

• Shopping list?
  • What sorts of behavior do shoppers exhibit?
  • What constraints are there on a shopper?
  • What improvements would make a better shopping list?
DESIGN DECISIONS

• Shopping list?

• Stack?
  • What sorts of behavior does the ‘stack’ support?
  • What constraints are there on a stack user? (Is there a change in certainty?)
  • What improvements would make a better stack? (What problems might arise in a stack?)
STACK ADT

- Important to know *exactly* what we expect from a stack.
  - Push(Object a) returns null; *(other options?)*
  - Pop() returns Object a: where a is the element on ‘top’ of the stack; also removes a from the stack
  - Top() returns Object a: where a is the element on ‘top’ of the stack without removing that element from the stack
  - How long will these operations take?

That depends on the Data Structure and Implementation
STACK ADT

• Array implementation
• Unique problems?

What if the array is full?
What if we alternate push() and pop()?
STACK ADT

- Array implementation
- Unique problems?
  - End of Array
- Unique solutions?
  - Resizing (costly!)
  - Circular Array (?)
- Why use at all?