CSE 326: Data Structures

James Fogarty

Spring 2009

CSE 326 Team

• Instructor:
  › James Fogarty, CSE 666
• TAs:
  › Eric McCambridge
  › Josh Barr
• Email:
  › cse326-instr@cs.washington.edu
• Office Hours:
  › TBA
CSE 326 Team

• Instructor:
  › James Fogarty, CSE 666
• TAs:
  › Eric McCambridge
  › Josh Barr
• Email:
  › cse326-instr@cs.washington.edu
• Office Hours:
  › TBA
Today’s Outline

• Introductions
• Homework 0
• Course Administrivia
• Project 1
• What is this Course About?

Homework 0: Bring to Class on Wednesday:

• Name
• Year (1,2,3,4,5,6,??)
• Hometown

• Interesting Fact or “What I did on my spring break”
Today’s Outline

• Introductions
• Homework 0
• Course Administrivia
• Project 1
• What is this Course About?

Course Information

Webpage:

http://www.cs.washington.edu/326

Communication

Instructors
› cse326-instr@cs.washington.edu
› (or our individual addresses)

Announcements
› cse326a_sp09@u.washington.edu
› (you are automatically subscribed @u)
› You are responsible for traffic on this list

Discussion
› Discussion board linked off webpage
› Use your real name and picture

Feedback Always Welcome
› See anonymous link on webpage
Course Mechanics

• Approximate Grading
  25% - Written Homework Assignments
  25% - Programming Assignments
  20% - Midterm Exam (in class, fifth week)
  25% - Final Exam
  5% - Best of Programming or Exams

Written homeworks

Written homeworks (8 total)
  › Due at the start of class
  › Assigned on Fridays, Due on Fridays
  › Intended to Cover Material through the Monday after it is Assigned

  › No late homeworks accepted
Projects

- Programming projects (3 total, with phases)
  - In Java
  - Eclipse encouraged
  - Turned in electronically
  - Can have one “late day” for extra 24 hours
    Must email TA in advance

- Work in teams only on explicit team projects
  - Appropriate discussions encouraged

Project/Homework Guides

On the website - note especially:
  - Gilligan’s Island rule applies.
  - Homeworks: Use pseudocode, not code. A human being is reading your homeworks.
    - See website for pseudocode example.
  - Projects: code is only 40% of your grade!
  - Spend time commenting your code as you write - it will help you be a better programmer.
Section

What happens there?
› Answer questions about current homework
› Previous homeworks returned and discussed
› Discuss the project (getting started, getting through it, answering questions)
› Finer points of Java and whatnot
› Reinforce lecture material

An Important Lecture Question
Homework

Reading in Weiss
Chapter 1 – (Review) Mathematics and Java
Chapter 2 – Algorithm Analysis
Chapter 3 – Lists, Stacks, & Queues

Today’s Outline

• Introductions
• Homework 0
• Course Administrivia
• Project 1
• What is this Course About?
Project 1

• Soundblaster! Reverse a song
• Implement a stack and a queue to make the “Reverse” program work
• **Read the website**
  › Detailed description of assignment
  › Detailed description of how programming projects are graded

Today’s Outline

• Introductions
• Homework 0
• Course Administrivia
• Project 1
• What is this Course About?
Class Overview

• Introduction to many of the basic data structures used in computer software
  › Understand the data structures
  › Analyze the algorithms that use them
  › Know when to apply them
• Practice design and analysis of data structures
• Practice using these data structures by writing programs
• Make the transformation from programmer to computer scientist

Goals

• You will understand
  › What the tools are for storing and processing common data types
  › Which tools are appropriate for which need
• So that you can
  › Make good design choices as a developer, project manager, or system customer
• You will be able to
  › Justify your design decisions via formal reasoning
  › Communicate about programs clearly and precisely
Data Structures

“Clever” ways to organize information in order to enable efficient computation

› What does it mean to be clever?

› What does it mean to be efficient?

Picking the best Data Structure for the job

• The data structure you pick needs to support the operations you need
• Ideally it supports the operations you will use most often in an efficient manner
• Examples of operations:
  › A List with operations insert and delete
  › A Stack with operations push and pop
Why So Many Data Structures?

**Ideal data structure:**
“fast”, “elegant”, memory efficient

Generates tensions:
› time vs. space
› performance vs. elegance
› generality vs. simplicity
› one operation’s performance vs. another’s

---

**Terminology**

- **Abstract Data Type (ADT)**
  › Mathematical description of an object with set of operations on the object. Useful building block.

- **Algorithm**
  › A high level, language independent, description of a step-by-step process

- **Data structure**
  › A specific family of algorithms for implementing an abstract data type.

- **Implementation of data structure**
  › A specific implementation in a specific language
Program Abstraction

Problem definition:

Algorithm:

Implementation:

Data Abstraction

Abstract Data Type (ADT):

Data Structure:

Implementation:
Steve’s view of CSE

• 100 level courses, some 300 level
  › How to do stuff

• This course
  › Really cool ways to do stuff

• 400 level courses
  › How to do really cool stuff

James’s view of 326

• This is a course about learning how to attack non-trivial problems
  › Problems where it matters how you do them

• Example:
How Do You Decide Which Examples are Positive?

How About Now?
How Do You Choose a Representative Set?

Don’t Forget Your Homework

1) **Homework 0**
   Bring to lecture on Wednesday

2) **Reading** in Weiss