
CSE 143 Computer Programming II

Welcome!
Course Overview and Administrative

6/21/2004

(c) University of Washington

00-1

Outline for Today

- Course Overview
- Goals
- Administrative details
- Workload and grading
- Resources
- Background

This information is largely included in today's handouts, and is on the web – no need to transcribe, but do take notes about important items!

6/21/2004

(c) University of Washington

00-2

Introductions

- **Instructor: Martin Dickey**
dickey@cs.washington.edu
Allen Center 640, office hours TBA
- **TA's: Xu Miao, Lincoln Ritter, Scott Schremmer, David Tran, Xu Miao**
cse143-ta@cs.washington.edu (goes to all TAs and the instructor)
- **Lab Consultants: once we get their hours worked out, we'll post a schedule**
- **Course administrator: Pim Lustig**
cse143-admin@cs.washington.edu
- **Everyone on the course staff:**
cse143-staff@cs.washington.edu (reaches entire staff)
- **You!**

6/21/2004

(c) University of Washington

00-3

Not Registered?

- **Still quite a few slots left! Tell your friends to sign up**
- **Non-matriculated students, grad students, registration problems – please see Pim Lustig (Allen Center main CSE office): PL@cs.washington.edu**

6/21/2004

(c) University of Washington

00-4

Are You Ready?

- Course is a direct continuation of CSE 142 Java
- **Must have a firm grasp of Java basics**
 - including classes, objects, statements, expressions, methods, parameters, arrays, JavaDoc, etc.
concepts and terminology as well as being able to use in programs
 - No systematic review
 - Look at old CSE 142 web pages – you should be able to handle those assignments and exams
- **What if you took the C version of CSE 142? Or took 142 elsewhere?**
 - Let's talk about that now
 - Not sure?
 - Sit in on both for a few days
 - Try the first 143 assignment
 - We'll help you switch to 142 if that's your decision

6/21/2004

(c) University of Washington

00-5

Java!

A modern approach to programming including

- Objects everywhere; classes, interfaces, polymorphism
- Exceptions
- Streams and networking support
- Garbage collection
- Specifications, design by contract support
- Rich set of standard libraries
- Documentation tools and standards, on-line library documentation
- We'll use Sun's Java SDK 1.4.2 (windows, *nix, OS X)
 - 1.3 will *not* do; 1.4.1 is OK if that's the latest you can get (OS X Jaguar)
 - J++ (Microsoft) will *not* do
 - Please update your software!
 - Details: *Computing at Home* page on course web



6/21/2004

(c) University of Washington

00-6

Content Overview (1)

Programming language and libraries

- Classes, class relationships (inheritance) interfaces, types
- Debugging and systematic testing (JUnit)
- Graphical user interfaces & event-driven programming (Swing, event handling, model-view-controller design)
- Exceptions
- Stream I/O and files
- Recursion

6/21/2004

(c) University of Washington

00-7

Content Overview (2)

Data structures and algorithms

- Lists, stacks, queues, trees, dictionaries
- Implementation techniques: arrays, linked data structures
- Comparing implementations: basic complexity theory
- Divide and conquer algorithms: sorting and searching
- **And Much Much More!**

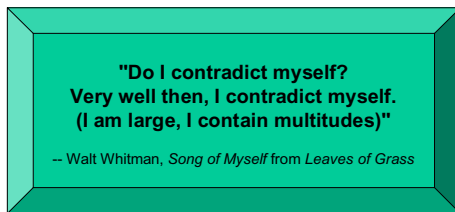
6/21/2004

(c) University of Washington

00-8

Course Objectives

- This *is* a programming course
- This is *not* a programming course



6/21/2004

(c) University of Washington

00-9

Is it or Isn't it?

- This *is* a programming course
 - The key goal is learning to program *well*, not just getting stuff to run
 - Good design, good organization, good style
 - Good algorithms, meaningful efficiency
- This is *not* a programming course
 - Lots of Java features won't be covered
 - See Java reference books & JavaDoc for full descriptions of the Java language & libraries
 - We cover the essential parts of Java that support good programming
 - Many important computer science topics
 - Some related to programming, but broader than Java
 - Data structures, algorithms, complexity analysis, software engineering...
- Fact: writing programs that work perfectly isn't enough to get a perfect grade (!)

6/21/2004

(c) University of Washington

00-10

My Goals for You

- Take you to the next technical step in programming
- Challenge you with material of considerable intellectual content, and with projects of considerable complexity.
- Develop your ability to learn independently
- Develop your ability to learn cooperatively
- Increase our awareness of larger issues surrounding the use of information technology in our world
- If possible, make it fun. If possible...

6/21/2004

(c) University of Washington

00-11

"Critical Thinking" Skills

- Reading with care
- Identifying incompleteness and ambiguity
 - and distinguishing between the two!
- Dealing with incompleteness and ambiguity
- Distinguishing important from unimportant issues
- Identifying and articulating assumptions
- Explaining, justifying, and supporting your conclusions

6/21/2004

(c) University of Washington

00-12

By the End of the Course...

- 5 things you should be able to do after CSE143
 - Be able to design and implement abstractions (classes) using modern programming language features and techniques
 - Be able to test and systematically locate and remove errors in programs
 - Be able to learn and use new libraries using standard documentation (no training wheels)
 - Be able to evaluate tradeoffs between different implementations of an abstraction and pick suitable ones
 - Be able to communicate technical concepts literately

6/21/2004

(c) University of Washington

00-13

My Expectations for You

- Responsibility
 - Keep up, know what's happening
 - Meet deadlines, budget your time, make backups
 - Take responsibility for your own code and debugging
- Respect
 - For others in the class (people sitting around you in lecture, members of your quiz section, partners on programming projects, ...)
 - For the course staff
 - For yourself

6/21/2004

(c) University of Washington

00-14

My Goals For Myself

- Top goals for the course:
 - Help all of you learn
 - Keep the course on track
 - Make the homework projects interesting
 - Make lecture and section events you look forward to!
 - Learn a *bunch* of names
 - Take a lot of pictures

6/21/2004

(c) University of Washington

00-15

Course Organization

- 3 lectures per week (MWF)
- Quiz section twice per week (Tu & Th)
 - Exercises, review, discussions, etc.
 - Extensive experience vs regular sections – section swaps
- Frequent quizzes
 - To keep you up with the reading and assignment instructions
 - To test mastery of current material
 - To provide TAs and me with feedback

6/21/2004

(c) University of Washington

00-16

What's Different About Summer

- Smaller class
- Hopefully more relaxed and personal
- No IPL
- Slightly compressed schedule
 - 9 weeks vs 10.5

6/21/2004

(c) University of Washington

00-17

Assignments

- Typically (but not always!) due Wed. night 9pm (electronic) and/or in sections or lecture Thursday or Friday morning (written)
 - Written assignments often collected all day in CSE office
- Primarily fairly substantial programming projects with written reports
- Maybe some shorter problems and programming drills
- Expect *something* to be due every week
- Assignments will more complex than in CSE142
 - Assignment directions, too!
- No late assignments accepted
 - But be sure to talk with your TA about problems truly beyond your control like illness or family emergency so we know what happened.

6/21/2004

(c) University of Washington

00-18



Academic (Mis)conduct



- Goal: balance the following
 - *Learning*: each student must do the work to learn effectively
 - *Cooperation*: people learn best when they can cooperate with others
 - *Fairness and honesty*: Nobody should ever represent the work of someone else as their own or try to claim credit for it

6/21/2004

(c) University of Washington

00-19



Academic (Mis)conduct



- Policy
 - You must do assignments by yourself or with your assigned partner (unless explicitly stated otherwise in an assignment)
 - You may discuss general approaches and ideas with others, but
 - You *may not ever* give code to or receive code from others
- We check this and act when trouble is discovered
- Use your common sense and ask first if unclear
 - Rule of thumb: *any activity you engage in for the purpose of earning credit while avoiding learning, or to help others do so, is likely to be an act of academic misconduct* (from CSE dept. policy – see link on the web)

6/21/2004

(c) University of Washington

00-20

Exams & Quizzes

- **Exams**
 - 2 midterm exams in class; probable dates: Friday week 3 and Friday week 6.
 - Final exam: Friday week 9: last day of class, regular time.
- The exams will not be given on any other days. Don't make plans which would take you away!
- Format: mixture of short answer, short essay, multiple choice, programming (both short and longer problems)

6/21/2004

(c) University of Washington

00-21

Grading

- **Grade distribution (subject to change)**
 - 30% homework assignments and projects
 - 16% + 16% midterm exams
 - 25% final exam
 - 8% quizzes
 - 5% participation, service, citizenship
- **Class is "curved"**
 - Median of final course grades is around 3.0
 - Maybe a bit higher when there are a lot of drops
 - Definitely higher if everyone does a great job (but statistically unlikely)
- **Why?**

6/21/2004

(c) University of Washington

00-22

Grading

- Project and quiz grading will be very coarse
- No partial points
- Typical scale: 4, 3, 2, 1, 0 for projects and written reports
 - Mastery || Good Job! || On the Right Track || Honest Effort, but... || Really, Now!
 - Intermediate turns typically 3, 2, 1, 0 (all well, some problems, serious problems, not credible)
 - Separate scores for program operation/code quality
 - i.e., Yes! Clarity, readability, style matters
 - Written reports count as much as the actual code (being able to communicate what you do is a crucial skill)
- Other assignments, typically 2, 1, 0 per question or question part
- Quiz question grading: check (1), check minus (1, but you should have been more on top of things, 0

6/21/2004

(c) University of Washington

00-23

Resources to Help You Succeed

- **Course staff**
 - Your TA is your primary contact, but please feel free to talk to any of us
 - Especially: don't leave me lonely in office hours!*
 - I'll try to be available right *after* class on Monday and Friday for as long as there are questions
 - but *before* class, it's panic time. Apologies in advance if I'm grouchy then.
 - **Consultants in the Lab**
 - A limited resource!
 - **No CLUE**
 - (An evening programming during the regular school year)

6/21/2004

(c) University of Washington

00-24

More Resources

- Help each other! Form study groups, spend time on the discussion list, etc.
- Undergraduate advisors, for general questions about the CSE programs (Allen Center)
- College of Engineering has some special resources for women and minorities
- Other university resources

6/21/2004

(c) University of Washington

00-25

For Reading and Study

- Lecture slides and course notes
 - **Alert!** Not all lecture material is on the slides!
 - Slides used will be posted on the web
NOT distributed in lecture
- Textbook: Next slide
- Other Material
 - Possibly handouts
 - All e-mail announcements, assignment descriptions, etc. should be considered required reading. They could even be tested on!

6/21/2004

(c) University of Washington

00-26

Textbooks

- Textbook: Niño & Hosch, *An Introduction to Programming and Object-Oriented Design using Java*, 2nd edition, 2004.
 - You can use the Spring 2004 Custom Edition if you can find it
 - Earlier editions are not advised
 - **Alert!** We may not follow the book very closely!
 - There will be reading assignments from this book.
If you choose not to buy it, be sure you have access to a copy
 - Will not always match our way of doing things, or our order!
But does at least provide a complementary view

6/21/2004

(c) University of Washington

00-27

Communicating Electronically

- Course web site
- Discussion Board: linked from Web site
 - UWNetID required
 - Open discussion – please contribute!
 - Course staff monitors and contributes as needed
- Email to us for things not appropriate for public discussion
 - Addresses on the web
 - Email works better for some things than other (e.g., very bad for trying to debug code)
- E-mail from us: cse143-announce
 - Sent directly to your UWNetID account
 - We'll try to keep the spam to a minimum, but... you must read what we do send!

6/21/2004

(c) University of Washington

00-28

Computing Facilities

- **Introductory Programming Lab (IPL) -- NOT OPEN IN SUMMER**
- **Other campus labs are available**
- **Computing at home**
 - **Java software and tools are freely available for download**
Java version MUST be 1.4+ Install entire SDK (Windows, Linux), or run software update (Mac OS X)
You're free to use any Java development environment
Recommended: Dr.Java (powerful but pretty simple), Eclipse (industrial strength)
See Computing At Home page for links and details
- **Even if you plan to compute at home, learn your way around the UW labs**

6/21/2004

(c) University of Washington

00-29

Your First CSE143 Assignment

- **Required reading:**
 - syllabus, academic conduct policy page.
 - Do this before quiz section tomorrow!
 - Either buy or make sure you have access to a textbook
 - Review whatever Java you learned in 142
- **Review rest of web (still somewhat incomplete)**
 - Find the first day's slides
 - Watch for the discussion board and announcements archive to become available
- **Install the needed software on your home computer**
 - and/or visit a campus lab and locate the software
- **(After tomorrow) memorize your quiz section # and TA's name**

6/21/2004

(c) University of Washington

00-30