

---

**CSE 142**  
**University of Washington**  
**Autumn 2003**

Welcome!  
Organization & Administration  
3 handouts today  
Syllabus, Calendar, and a first Assignment

10/1/2003

(c) 2001-2003, University of Washington

A-1

---

**Outline for Today**

- Course Overview
  - Administrative details
  - Workload and Resources
  - Work submission and Grading policies
  - And a brief introduction to computer science & modeling
- 
- This information (and more) is included in today's handouts, and is on the web – no need to transcribe; just note highlights
  - Some things are new or different this quarter – be sure you're using current information

10/1/2003

(c) 2001-2003, University of Washington

A-2

---

**Introductions**

- **Instructors**
  - Martin Dickey (9:30) & Rob Duisberg (11:30)  
cse142-instructors@cs.washington.edu
- **TAs**
  - Many – see next slide  
cse142-tas@cs.washington.edu
- **Course Administrator**
  - Pim Lustig  
cse142-admin@cs.washington.edu
- **Consultants:** Savvy students we've hired to help out in the lab  
cse142-staff@cs.washington.edu reaches entire staff
- **Students: You!**



10/1/2003

(c) 2001-2003, University of Washington

A-3

---

**Teaching Assistants**

- |                     |             |
|---------------------|-------------|
| • Shaverdian Arpi   | arpi@cs     |
| • Orion Bawdon      | orion@cs    |
| • Gaurav Bhaya      | gbhaya@cs   |
| • Lillian Kittredge | kittredl@cs |
| • Lincoln Ritter    | lritter@cs  |
| • Jeremy Lingmann   | lingmann@cs |
| • Christopher Re    | chrisre@cs  |
| • Scott Schremmer   | scotths@cs  |
| • David Tran        | davidt23@cs |
| • Xu Miao           | xm@cs       |



10/1/2003

(c) 2001-2003, University of Washington

A-4

## Course Organization

- 3 lectures per week (MWF)
- Quiz section once per week (Thursday)
  - Regular quizzes (easy to do if you keep up)
  - Exercises, review, discussions, etc.  
Small groups of students will often work together on activities
- Designated quiz sections: more later
  - Regular
  - High-background?
  - Low-background?
- Informal gatherings and presentations evenings 7-10pm in Mary Gates Hall

10/1/2003

(c) 2001-2003, University of Washington

A-5

## Course Goals

- Learn general principles and practices of computer programming
- Develop programming skills in the context of Computer Science
  - Reading and Analysis
  - Design
  - Implementation
  - Writing and Documentation
  - Testing
  - Debugging
- Develop technical communication skills
  - This is hard – and important to do well
- (And learn some Java in the process)
- (and have some fun)

10/1/2003

(c) 2001-2003, University of Washington

A-6

## 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
- Develop your ability to deal with incomplete and ambiguous information
- Increase our awareness of larger issues surrounding the use of information technology in our world
- If possible, make it fun. If possible...

10/1/2003

(c) 2001-2003, University of Washington

A-7

## 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!
- Plus some more personal goals...
  - Learn some more Java myself
  - Make better use of technology in the classroom
  - Refine some teaching techniques
  - Take lots of pictures
  - And... learn a bunch of names!

10/1/2003

(c) 2001-2003, University of Washington

A-8

## Programming

- **Both easier and harder than most people make it out to be**
  - **Easier:** Many of the things good programmers do well are things that we already do all the time, but we don't think consciously about it
  - **Harder: Programming is in large part a skill or an art**
    - Requires a level of design, problem-solving, and precision that is not common in most of the rest of life
    - Rather like chess, or composing music: a process of creating abstract, dynamic structures and interactions
    - Very different from using applications or writing simple scripts
- **Best learned by practice, trying things out, and reasoning**
  - Don't worry - you won't break the computer by trying something new

10/1/2003

(c) 2001-2003, University of Washington

A-9

## 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
- If none of the above makes sense... don't worry! It will eventually
- We'll use Sun's Java SDK 1.4.2
  - 1.3 will *not* do.
  - J++ (Microsoft) will *not* do
  - Details: *Computing at Home* page



10/1/2003

(c) 2001-2003, University of Washington

A-10

## What to Expect

- **Homework assignments**
  - Frequent but irregular schedule
  - Mix of written problems and short programming exercises, some using a computer
  - Done individually
- **Longer programming projects**
  - 3-4 of these
  - Up to 2 weeks each
  - Work with a partner - pair programming
    - Partners assigned by course staff; different partner for each project
  - Individual written reports for each project
- **Discussions and activities in lectures and quiz sections**
- **Designated textbook sections**
- **Reading carefully and following instructions are key to success in this course**

10/1/2003

(c) 2001-2003, University of Washington

A-11

## 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 for full descriptions of the Java language
  - We emphasize features of Java that support good programming
- Many important computer science topics
  - Some related to programming, but broader than Java
  - Data types, structures, algorithms, complexity analysis, software engineering...
- **Fact:: writing programs that work perfectly isn't enough to get a perfect grade (!)**

10/1/2003

(c) 2001-2003, University of Washington

A-12

## Who is the Course For?

- Course is for beginners, who...
  - Want a serious and rigorous introduction to programming and computer science
  - Can commit to the effort needed to succeed
- Previous programming experience is *not* a prerequisite!
- You should be comfortable with Math, Science, and English through the 12<sup>th</sup> grade level
- If you have already programmed...
  - In Java or C++? Did pretty well? Consider going right on to CSE143  
Lecture MWF 11:30 pm Gugg 224 – try it *today!* (if you're in the 9:30 section...)
  - If you are not a beginner: remember that the course is *not* primarily for you  
If you stay, please participate and be helpful

10/1/2003

(c) 2001-2003, University of Washington

A-13

## Keeping Up

- Course is for beginners, however...
- Material is cumulative
  - *Essential* to keep up
  - Ask for help the moment you need it; don't fall behind
- *No late assignments accepted*; no makeup exams or quizzes – need to keep on schedule
- Talk to course staff and fellow students
  - We're here to help
  - But ultimately it's up to you

"I waited for hours for the consultant" is no excuse – figure it out yourself!!

10/1/2003

(c) 2001-2003, University of Washington

A-14

## Communication

- People learn best when they ask questions and discuss material
  - With each other, with course staff, with friends, both in and out of class  
Ask questions; participate!
  - The informal evening sessions with staff in MGH may be ideal for this!
- Main discussion channel: EPost Message Board
  - Link on course web page
  - Read this regularly & contribute when you can
  - Course staff will participate and contribute
  - You *must* use the Message Board as the starting point for technical questions
  - You *may not* post code to the Message Board



10/1/2003

(c) 2001-2003, University of Washington

A-15

## Resources to Help You Succeed

- Course staff
  - We're all in this together – feel free to talk to *any* TA or instructor and come to *anyone's* office hours
  - Use email to set up appointments at other times if office hours don't work.
- Main information source: course web pages
  - [www.cs.washington.edu/142](http://www.cs.washington.edu/142)
  - Start browsing now – be sure you can find your way around
- [cse142-announce@cs](mailto:cse142-announce@cs.washington.edu) mailing list for urgent messages from CSE142 staff to everyone
  - Registered students are included on this list automatically
- Staff email addresses for things that are not appropriate for the discussion board – details on the course web

10/1/2003

(c) 2001-2003, University of Washington

A-16

## Book and Lecture Slides

- **Textbook: *An Introduction to Programming and Object-Oriented Design* by Nino & Hosch –**
  - Special new edition, available in U. Bookstore
    - Can *not* be ordered otherwise (e.g., amazon.com)
  - See **course calendar** for readings to do before class,  
(latest version on the course Calendar page)
- **Updated lecture slides will be posted to the course web, sometime after the topic is completed**
  - You can print the preliminary version, look at it before lecture, and bring it with you to take notes
  - Lecture slides are **not** a substitute for attending class!
    - there will be additional information, explanations, and activities in class that do not appear on the printed slides

10/1/2003

(c) 2001-2003, University of Washington

A-17

## Assessment

- **Short mini-quizzes in class (regularly)**
  - Graded on a simple scale
  - cover current readings and recent topics, whether discussed in class or not
- **Midterm exams in lecture**
  - Friday, October 24 and Friday, November 14 (tentative, but likely)
- **Final exam**
  - Tuesday, December 16<sup>th</sup>.
  - **You must take the final exam on Tuesday, Dec 16<sup>th</sup>– do not plan to leave campus early**
    - No matter how good a discount airfare you can get on Dec 15<sup>th</sup>!
- **Exams are a mix of multiple choice, written questions, short programming problems, etc.**
- **Exams do not necessarily assess the same skills and knowledge as the projects and homework!**

10/1/2003

(c) 2001-2003, University of Washington

A-18

## Disconnect?

- The parts of the course have different goals and styles
  - May seem disconnected from one another
- **Tests vs. projects**
  - Each measures things that the other can't
  - Tests may seem hard even when homework doesn't!
  - Homework may require learning about topics not covered in lecture
- **Lectures vs. homework**
  - Lectures may cover topics not practiced in homework
  - Lectures cover concepts and examples; will rarely talk about homework
  - Lectures sometimes mathematical, homework rarely so
- **Quiz sections**
  - active learning, practice, and review of recent topics



10/1/2003

(c) 2001-2003, University of Washington

A-19

## Grading

- **Anticipated breakdown**
  - 35% Homework and projects
  - 14% + 16% Midterm exams
  - 21% Final exam
  - 10% Quizzes
    - weighted equally, regardless of length or difficulty
  - 4% Service and participation
    - in-class activities, class participation, assistance to class members and staff, taking initiative to host evenings or speakers at MGH, etc.
- **Individual assignments and projects may weighted differently**
  - depending on difficulty, length, etc.
- **Percentage breakdown may change a fraction**
  - depending on how the course evolves over the quarter

10/1/2003

(c) 2001-2003, University of Washington

A-20

## Collaboration vs Academic Misconduct

- While you should discuss ideas and learn with others, it is academic misconduct to represent someone else's work as your own, even if you have modified it
  - Same standard as in an English or History class – nothing changes because computer code might be involved
- You should acknowledge places where you receive help on homework or projects
  - "Help" means discussing problems, getting suggestions, but not writing up actual solutions or code (except with partner on programming projects)
- We have sophisticated software tools to check for problems, and we follow up when we find them
  - You *don't* want to receive an invitation to meet with the Vice Provost

10/1/2003

(c) 2001-2003, University of Washington

A-21

## More About Quiz Sections

- Regular: designed for all students – no prior experience
- High-background: designed for students with prior exposure to computing – chance to go into additional technical details, etc.
- All sections have the same assignments, take the same tests, and are graded the same
- On Wednesday, you may be able to request a switch to a different kind of section – we'll do the best we can to accommodate preferences
  - Between now and then, find out which section you're registered for and what kind it is
- Possible to informally switch sections with permission of TAs involved, even after Wednesday – no registration change needed

10/1/2003

(c) 2001-2003, University of Washington

A-22

## Computing Facilities

- CSE142 uses the UWired general labs
- Primary lab for CSE142/143 is the Introductory Programming Lab (IPL), 3<sup>rd</sup> floor Mary Gates Hall (MGH)
  - Pay a visit there today!
  - Course consulting staff available in the IPL
  - Can also use machines in Computing Commons in MGH and Odegaard (OUGL)
- Computing at home
  - Course software and tools are freely available for download
  - Instructions on the CSE 142 web
- Many assignments are submitted via the web
  - Very important to follow *exactly* the instructions for turning in each assignment!
  - You don't follow the instructions – you don't get credit

10/1/2003

(c) 2001-2003, University of Washington

A-23

## Can't Get In?

- You're welcome to attend today anyway!
- New slots open up as people drop
- No waiting list
- No entry codes
- Attend lectures and any old quiz section for the time being. But no guarantees – you might not get in.
- If you aren't registered by Wednesday or so – consider making a new plan



10/1/2003

(c) 2001-2003, University of Washington

A-24