CSc 115/160 (S01)

Fundamentals of Computer Programming II

Spring 2002

Introduction

- Web page: http://www.csc.uvic.ca/~csc115/
- Your instructors:
  - S01
    - Hausi Muller, EOW 337, hausi@csr.uvic.ca
    - Office hours: MR 2:00 - 3:00 p.m., W 1:00 - 2:00 p.m., or by appointment
  - S02
    - Bette Bultena, ELW B216, bbultena@csc.uvic.ca
    - Office Hours : M 11:00 - 12:00 a.m., T 10:30 - 11:30 p.m, F 2:00 - 3:00 p.m., or by appointment
- Required Text
  - *Data Structures and Algorithms in Java Second Edition*, by Michael T. Goodrich and Roberto Tamassia, published by John Wiley & Sons, 2001
- Course objectives
  - Introduction to fundamental data structures and algorithms
  - Fundamentals of object-oriented programming

Laboratories

- Lectures
  - Attending lectures is mandatory
  - Essential for doing well on all exams
- Labs
  - Start week of Jan 14
  - One hour of labs a week
  - Attendance is required
  - Make sure you attend the section you are signed up for
  - Extra details and hints on assignments
- Lab leader:
  - Mike Easton (csc115@csc.uvic.ca)
- Mark coordinator:
  - TBA
- Lab instructors:
  - Dylan Dawson

Evaluation

Assignments (5, 4% each)	20%
Midterms (2, 20% each)	40%
Final (must-pass)	40%

- Marks will be posted publicly on the web by student number
  - if you don't want your marks posted in this manner, notify me by email before January 18
- The Midterms will be one hour, closed books, closed notes, no calculators, in class Feb 6/7 and Mar 6/7.
- Final exam will be three hours, closed books, closed notes, no calculators, scheduled by the registrar.

Calendar

Due Dates	Assignments
Tue, Jan 29	Assignment 1
Tue, Feb 12	Assignment 2
Wed/Thu, Feb 6/7	1st Midterm Exam
Tue, Mar 5	Assignment 3
Wed/Thu, Mar 6/7	2nd Midterm Exam
Tue, Mar 19	Assignment 4
Tue, Apr 2	Assignment 5
TBA	Final Exam

Assignments

- 5 assignments during the course
  - ➢ evenly weighted, 4% each
- Late submissions are not accepted
  - ➢ if valid excuse (e.g., doctor's statement), raise weight of other assignments to compensate
- Work in the labs or at home
  - ➢ use your favorite Java environment, JDK 1.2
  - ➢ IBM VisualAge Java 3.5 (recommended)
    - Entry version is free
    - http://www-4.ibm.com/software/ad/vajava/
    - http://www7.software.ibm.com/vad.nsf/Data/Document4293
  - ➢ we only support the JDK and CodeWarrior on university systems
- Cheating: zero-tolerance policy
  - ➢ first time fail assignment, second time fail course
  - ➢ discussion encouraged, but acknowledgements required

Prerequisites and Resources

- Everybody should have taken CSc 110 or a similar course
  - ➢ Basic Java knowledge and programming skills are assumed
  - ➢ Knowledge of object-based programming is assumed
- Resources (consult in this order):
  - ➢ Lab instructor
  - ➢ Computer Science Consulting Office (ELW B210, 721-7204, helpme@csc.uvic.ca)
  - ➢ Course web page (http://www.csc.uvic.ca/~csc115/)
  - ➢ Computer labs web page (http://www.csc.uvic.ca/~labspg/)
  - ➢ Your course instructors

Departmental Course Outline (May 1996)

- Variables (12%)
  - ➢ Extent and scope
  - ➢ Local, static, and global variables
  - ➢ Pointers and arrays
  - ➢ Dynamic storage allocation
  - ➢ Parameter mechanisms
  - ➢ Constants
- Object-oriented programming (20%)
  - ➢ Encapsulation, data abstraction
  - ➢ User-defined data types, classes
  - ➢ Interface vs implementation
  - ➢ Data members
  - ➢ Member functions (methods)
  - ➢ Constructors
  - ➢ Derived classes, inheritance
- Fundamental data structures (20%)
  - ➢ Stacks, queues, linked lists
  - ➢ Binary trees
- Recursion (6%)
- Algorithms (10%)
  - ➢ Analysis, searching/sorting
- Reasoning about programs (6%)
  - ➢ Assertions, loop invariants
  - ➢ Pre- and post-conditions
- Separate compilation (3%)
  - ➢ Modularization
  - ➢ Multi-file programs
  - ➢ Header files, conditional compilation, the preprocessor
- Use of libraries (3%)
  - ➢ Standard libraries
  - ➢ Input and output
  - ➢ String manipulation
- Discretionary material (20%)
  - ➢ Exceptions
  - ➢ Unified Modeling Language (UML)
  - ➢ Lists, priority queues
  - ➢ Java ADT libraries