# COMP322-Spring-2012

***NOTE: the most recent web site for COMP 322 is located here***

| Home | Office Hours | Turnin Guide | HJ Info | Piazza | Resources |
|---|---|---|---|---|---|

## COMP 322: Fundamentals of Parallel Programming (Spring 2012)

| | | | |
|---|---|---|---|
| **Instructor:** | Prof. Vivek Sarkar, DH 3131 | **Graduate TA:** | Sanjay Chatterjee |
| | Please send all emails to comp322-staff at rice dot edu | **Graduate TA:** | Deepak Majeti |
| **Assistant:** | Amanda Nokleby, akn3@rice.edu, DH 3137 | **Graduate TA:** | Dragos Sbirlea |
| | | **Undergrad TA:** | Max Grossman |
| | | **Undergrad TA:** | Damien Stone |
| **Cross-listing:** | ELEC 323 | **Undergrad TA:** | Yunming Zhang |
| | | **Research Programmer:** | Vincent Cavé |
| **Lectures:** | **Brockman 101 (new location effective 1/18/2012)** | **Lecture times:** | MWF 1:00 - 1:50pm |
| **Labs:** | Ryon 102 | **Lab times:** | Tuesday, 4:00 - 5:20pm (Section 3) |
| | | | Wednesday, 3:30 - 4:50pm (Section 2) |
| | | | Thursday, 4:00 - 5:20pm (Section 1) |

### Introduction

The goal of COMP 322 is to introduce you to the fundamentals of parallel programming and parallel algorithms, using a pedagogical approach that exposes you to the intellectual challenges in parallel software without enmeshing you in low-level details of different parallel systems.  To that end, the main pre-requisite course requirement is COMP 215 or equivalent.  This course should be accessible to anyone familiar with the foundations of sequential algorithms and data structures, and with basic Java programming.  COMP 221 is also recommended as a co-requisite.

The pedagogical approach will introduce you to the following foundations of parallel programming:

- Primitive constructs for task creation & termination, collective & point-to-point synchronization, task and data distribution, and data parallelism
- Abstract models of parallel computations and computation graphs
- Parallel algorithms and data structures including lists, strings, trees, graphs, matrices
- Common parallel programming patterns including task parallelism, undirected and directed synchronization, data parallelism, divide-and-conquer parallelism, map-reduce, concurrent event processing including graphical user interfaces.

Laboratory assignments will explore these topics through a simple parallel extension to the Java language called Habanero-Java (HJ), developed in the Habanero Multicore Software Research project at Rice University.  The use of Java will be confined to a subset of the Java 1.4 language that should also be accessible to C programmers --- no advanced Java features (e.g., generics) will be used.  An abstract performance model for HJ programs will be available to aid you in complexity analysis of parallel programs before you embark on performance evaluations on real parallel machines.  We will conclude the course by introducing you to some real-world parallel programming models including the Java Concurrency Utilities, Google's MapReduce, CUDA and MPI.  The foundations gained in this course will prepare you for advanced courses on Parallel Computing offered at Rice (COMP 422, COMP 522).

Since the aim of the course is for you to gain both theoretical and practical knowledge of the foundations of parallel programming, the weightage for course work will be balanced across homeworks, exams, and lab attendance.

## Textbooks

There are no required textbooks for the class. You will be expected to read each lecture handout before coming to the lecture.  We will also provide a number of references in the slides and handouts.

However, there are a few optional textbooks that we will draw from quite heavily.  You are encouraged to get copies of any or all of these books.  They will serve as useful references both during and after this course:

- Java Concurrency in Practice by Brian Goetz with Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea
- Principles of Parallel Programming by Calvin Lin and Lawrence Snyder
- The Art of Multiprocessor Programming by Maurice Herlihy and Nir Shavit

## Lecture Schedule

| | Day | Date (2012) | Topic | Slides | Audio (Panopto) | Code Examples | Homework Assigned | Homework Due |
|---|---|---|---|---|---|---|---|---|
| **1** | Mon | Jan 9 | Lecture 1: The What and Why of Parallel Programming | lec1-slides | | ArraySum0.hj | HW1 (Written Assignment) | |
| **2** | Wed | Jan 11 | Lecture 2: Async-Finish Parallel Programming and Computation Graphs | lec2-slides | lec2-audio | PrimeSieve.hj | | |
| **3** | Fri | Jan 13 | Lecture 3: Computation Graphs, Abstract Performance Metrics, Array Reductions | lec3-slides | lec3-audio | ArraySum1.hj | HW2 (HJ Programming Assignment) | HW1 |
| **-** | Mon | Jan 16 | School Holiday | | | | | |
| **4** | Wed | Jan 18 | Lecture 4: Parallel Speedup, Efficiency, Amdahl's Law | lec4-slides | lec4-audio | | | |
| **5** | Fri | Jan 20 | Lecture 5: Data & Control Flow with Async Tasks, Data Races | lec5-slides | lec5-audio | (See Lab 3) | | |
| **6** | Mon | Jan 23 | Lecture 6: Memory Models, Atomic Variables | lec6-slides | lec6-audio | (See Lab 3) | | |
| **7** | Wed | Jan 25 | Lecture 7: Memory Models (contd), Futures --- Tasks with Return Values | lec7-slides | lec7-audio | ArraySum2.hj | | |
| **8** | Fri | Jan 27 | Lecture 8: Futures (contd), Dataflow Programming, Data-Driven Tasks | lec8-slides | lec8-audio | binarytrees.hj | | |
| **9** | Mon | Jan 30 | Lecture 9: Abstract vs. Real Performance, seq clause, forasync loops | lec9-slides | lec9-audio | nqueens.hj | | HW2 |
| **10** | Wed | Feb 01 | Lecture 10: Forasync Chunking, Parallel Prefix Sum algorithm | lec10-slides | lec10-audio | | | |
| **11** | Fri | Feb 03 | Lecture 11: Parallel Prefix Sum (contd), Parallel Quicksort | lec11-slides | lec11-audio | | HW3 (HJ Programming Assignment), SeqScoring.hj, X.txt, Y.txt, BigSeq.zip | |
| **12** | Mon | Feb 06 | Lecture 12: Finish Accumulators, Forall Loops and Barrier Synchronization | lec12-slides | lec12-audio | | | |
| **13** | Wed | Feb 08 | Lecture 13: Forall Loops and Barrier Synchronization (contd) | lec13-slides | lec13-audio | | | |
| **14** | Fri | Feb 10 | Lecture 14: Point-to-point Synchronization and Phasers | lec14-slides | lec14-audio | | | |
| **15** | Mon | Feb 13 | Lecture 15: Phaser Accumulators, Bounded Phasers | lec15-slides | lec15-audio | | | |
| **16** | Wed | Feb 15 | Lecture 16: Summary of Barriers and Phasers | lec16-slides | lec16-audio | | | |
| **17** | Fri | Feb 17 | Lecture 17: Task Affinity with Places | lec17-slides | lec17-audio | | | |
| **18** | Mon | Feb 20 | Lecture 18: Task Affinity with Places (contd) | lec18-slides | lec18-audio | | | |
| **19** | Wed | Feb 22 | Lecture 19: Midterm Summary | lec19-slides | | | | |
| **-** | F | Feb 24 | No Lecture (Take-home Exam 1 due by 4pm today) | | | | | HW3 |
| **-** | M-F | Feb 27 - Mar 02 | Spring Break | | | | | |
| **20** | Mon | Mar 05 | Lecture 20: Critical sections and the Isolated statement | lec20-slides | lec20-audio | | | |
| **21** | Wed | Mar 07 | Lecture 21: Isolated statement (contd), Monitors, Actors | lec21-slides | lec21-audio | | HW4 (HJ Programming Assignment), hw_4.zip | |
| **22** | Fri | Mar 09 | Lecture 22: Actors (contd) | lec22-slides | lec22-audio | HJ Actor Examples | | |
| **23** | Mon | Mar 12 | Lecture 23: Linearizability of Concurrent Objects | lec23-slides | lec23-audio | | | |
| **24** | Wed | Mar 14 | Lecture 24: Linearizability of Concurrent Objects (contd) | lec24-slides | lec24-audio | | | |
| **25** | Fri | Mar 16 | Lecture 25: Safety and Liveness Properties | lec25-slides | lec25-audio | | | |
| **26** | Mon | Mar 19 | Lecture 26: Parallel Programming Patterns | lec26-slides | lec26-audio | | | |
| **27** | Wed | Mar 21 | Lecture 27: Introduction to Java Threads | lec27-slides | lec27-audio | | HW5 (Written Assignment) --- HW5.pdf or HW5.doc | HW4 |
| **-** | Fri | Mar 23 | Midterm Recess | | | | | |
| **28** | Mon | Mar 26 | Lecture 28: Bitonic Sort (guest lecture by Prof. John Mellor-Crummey) | lec28-slides | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **29** | Wed | Mar 28 | Lecture 29: Java Threads (contd), Java synchronized statement | lec29-slides | lec29-audio | | | |
| **30** | Fri | Mar 30 | Lecture 30: Java synchronized statement (contd), advanced locking | lec30-slides | lec30-audio | | | |
| **31** | Mon | Apr 02 | Lecture 31: Java Executors and Synchronizers | lec31-slides | lec31-audio | | | |
| **32** | Wed | Apr 04 | Lecture 32: Volatile Variables and Java Memory Model | lec32-slides | lec32-audio | | | |
| **33** | Fri | Apr 06 | Lecture 33: Message Passing Interface (MPI) | lec33-slides | lec33-audio | | | HW5 |
| **34** | Mon | Apr 09 | Lecture 34: Message Passing Interface (MPI, contd) | lec34-slides | lec34-audio | | HW6 (Java Programming Assignment) , hw_6.zip | |
| **35** | Wed | Apr 11 | Lecture 35: Cloud Computing, Map Reduce | lec35-slides | lec35-audio | | | |
| **36** | Fri | Apr 13 | Lecture 36: Map Reduce (contd) | lec36-slides | lec36-audio | | | |
| **37** | Mon | Apr 16 | Lecture 37: Speculative parallelization of isolated blocks (Guest lecture by Prof. Swarat Chaudhuri) | lec37-slides | | | | |
| **38** | Wed | Apr 18 | Lecture 38: Comparison of Parallel Programming Models | lec38-slides | lec38-audio | | | |
| **39** | Fri | Apr 20 | Lecture 39: Course Review | lec39-slides | lec39-audio | | Exam 2 (Take-home) | HW6 |
| **-** | Fri | Apr 27 | Exam 2 due | | | | | Exam 2 |

## Lab Schedule

| Lab # | Date (2011) | Topic | Handouts | Code Examples | Solutions |
|---|---|---|---|---|---|
| **1** | Jan 10, 11, 12 | DrHJ setup, Async-Finish Parallel Programming | lab1-handout | HelloWorld.hj, ReciprocalArraySum.hj, PrimeSieve.hj | |
| **2** | Jan 17, 18, 19 | Abstract performance metrics with async & finish | lab2-handout | Search.hj | |
| **3** | Jan 23, 25, 26 | Data race detection and repair | lab3-handout | RacyArraySum1.hj, RacyFib.hj, RacyNQueens.hj, RacyFannkuch.hj | |
| **4** | Jan 30 Feb 01, 02 | Real performance, work-sharing and work-stealing runtimes, futures | lab4-handout | nqueens.hj, ArraySum2.hj | |
| **5** | Feb 07, 08, 09 | Data-driven futures | lab5-handout | MatrixEval.hj, test0.txt, test.txt, DDFEx.hj | |
| **6** | Feb 14, 15, 16 | Barriers and Phasers | lab6-handout | OneDimAveraging.hj | |
| **-** | Feb 21, 22, 23 | No lab (Exam 1 week) | | | |
| **7** | Mar 06, 07, 08 | Atomic Variables and Isolated Statement | lab7-handout | spanning_tree_isolated.hj, SortedListExampleGbl.hj | |
| **8** | Mar 13, 14, 15 | Actors | lab8-handout | HJ Actor Examples | |
| **-** | Mar 20, 21, 22 | No lab (HW4 deadline, midterm recess) | | | |
| **9** | Mar 27, 28, 29 | Java Threads | lab9-handout | nqueens.hj spanning_tree_atomic.hj | |
| **10** | Apr 03, 04, 05 | Java Locks | lab10-handout | lab10.zip | |
| **11** | Apr 10, 11, 12 | Message Passing Interface (MPI) | lab11-handout | lab11.zip | |
| **12** | Apr 17, 18, 19 | Map Reduce | lab12-handout | WordCount.hj  MapReduce.hj, words.txt Index.hj | |

## Grading, Honor Code Policy, Processes and Procedures

Grading will be based on your performance on six homeworks (worth 50%), two exams (20% each), and lab attendance (10%).

The purpose of the homeworks is to train you to solve problems and to help deepen your understanding of concepts introduced in class. Homeworks and programming assignments are due on the dates and times specified in the course schedule. Please turn in all your homeworks using the CLEAR turn-in system. Homework is worth full credit when turned in on time. A 10% penalty per day will be levied on late homeworks, up to a maximum of 6 days. No submissions will be accepted more than 6 days after the due date.

You will be expected to follow the Honor Code in all homeworks and exams. All submitted homeworks are expected to be the result of your individual effort. You are free to discuss course material and approaches to problems with your other classmates, the teaching assistants and the professor, but you should never misrepresent someone else's work as your own. If you use any material from external sources, you must provide proper attribution ([as shown here|http://www.dartmouth.edu/~writing/sources/]). Exams 1 and 2, which are pledged under the Honor Code, test your individual understanding and knowledge of the material. Collaboration on exams is strictly forbidden. Finally, it is also your responsibility to protect your homeworks and exams from unauthorized access.

Graded homeworks will be returned to you via email, and exams as marked-up hardcopies. If you believe we have made an error in grading your homework or exam, please bring the matter to our attention within one week.

## Past Offerings of COMP 322

- Spring 2011
- Fall 2009

## Accommodations for Students with Special Needs

Students with disabilities are encouraged to contact me during the first two weeks of class regarding any special needs. Students with disabilities should also contact Disabled Student Services in the Ley Student Center and the Rice Disability Support Services.