

# INFOSYS 255: Foundations of Software Design

## General Information

### Table of contents

1. Class Description.....	1
2. Instructor.....	1
3. Tutors (a.k.a. "Teaching Assistants").....	2
4. Location and Time.....	2
5. Organization.....	2
6. Books.....	2
7. Course Prerequisites.....	3
8. Workload.....	3
9. Getting Help.....	4
10. Grading.....	4
11. Testing Out of the Course.....	4
12. Java Tutors.....	5
13. Academic Honesty.....	5

### 1. Class Description

This class is an intensive introduction to programming principles and practice that prepares students with a non-technical background for the more technical SIMS courses. Topics include the fundamentals of object-oriented design, standard data structures and algorithms, software development, and computer program execution. This class is not simply a second semester course in the use of the Java programming language.

### 2. Instructor

[Brian S. Hayes](#)

Email: [bhayes@sims.berkeley.edu](mailto:bhayes@sims.berkeley.edu)

Office Location: South Hall, room 303B

Office Hours: Tuesdays, 3:00 pm to 4:30 pm, **or** by appointment.

### 3. Tutors (a.k.a. "Teaching Assistants")

The class tutors are here to help the students with the understanding of the course material and with the assignments. The tutors are responsible for leading the laboratory portion of the course. The class tutors are not intended to be tutors for understanding the basics of Java. If a student needs a [Java tutor](#) they are responsible for arranging for one.

**Lisa de Larios-Heiman**

Office Hours: Wednesdays, 11:00 am to 12:00 pm

Office Location: South Hall, Lab, room 210

**Jesse Mendelsohn**

Office Hours: Tuesdays, 4:00 pm to 5:00 pm

Office Location: South Hall, Lab, room 210

**Brooke Maury**

Office Hours: Wednesdays, 3:30 pm to 4:30 pm

Office Location: South Hall, Lab, room 210

Check the faces page to see what they look like!

### 4. Location and Time

The primary lecture meets Tuesday and Thursday from 9:00 am to 10:30 am in 202 South Hall. The required lab meets in South Hall, room 210, from 2:00 pm to 3:30 pm in the South Hall second floor computer lab, room 210.

### 5. Organization

This is a lecture-laboratory course where the lecture is primarily used to discuss the various topics relevant software design. The lab time is used for a number of different purposes including small programming projects relevant to lecture topics, getting instruction on software development tools, assistance with assignments, and introduction to the SIMS computing environment. In some cases, the lecture may include some lab-like activities.

### 6. Books

**Head First Java**

*General Information*

By Kathy Sierra & Bert Bates.

Publisher: O'Reilly & Associates; 1st edition (May 1, 2003).

ISBN: 0596004656

**Designing Object-Oriented Software**

By Rebecca Wirfs-Brock, Brian Wilkerson, Lauren Wiener.

Publisher: Pearson Education; (June 18, 1990)

ISBN: 0136298257

In the past, students were able to support the SIMS Information Management Students Association (IMSA) by purchasing books via the IMSA web page.

## **7. Course Prerequisites**

Students are expected to be knowledgeable of the basics of Java programming. This is discussed in the Programming Competency section of the Masters Admissions Requirements and is further detailed in the incoming student's Programming Skills message.

At a minimum, students should have the equivalent experience of an introductory programming class in Java. Students that do not know how to program, or have very little programming experience, are strongly encouraged to take at least one Java programming class before this class starts. Students should reread the Programming Skills message to see what previous INFOSYS 255 students have written about this. A poor understanding of Java fundamentals (e.g. syntax, writing classes, using objects, etc.) may dramatically increase the amount of time a student spends on this class and may get in the way of experiencing important aspects of the assignments.

Students should have enough Java knowledge and experience to write a simple calculator program where a user can add, subtract, multiply, and divide numbers via a command line interface. The program should have at least a few methods and include error handling.

Consent of instructor required for non SIMS students.

## **8. Workload**

The course evaluations from Fall 2003 indicated that students were spending 8 to 40 hours per week, including lecture and lab attendance, with an average of about 15 hours per week. Students might put in fewer hours during weeks when an assignment is not due. The amount of time spent will depend on the student's experience with programming, general aptitude for programming, and tendency to put off starting assignments. Of course, the instructor's ability and the teaching assistants' ability to educate you will have some bearing on this!

There will be readings each week. Students are expected to do the readings before lecture. Each week there will also be a required lab session. Often the lab will consist of exercises to

better learn the material from class.

## 9. Getting Help

You are encouraged to ask questions during lectures, office hours, the lab, and via email. If the scheduled office hours are not convenient for you, please make an appointment for a more suitable time.

Your email options are:

[is255@sims.berkeley.edu](mailto:is255@sims.berkeley.edu)

This is the class mailing list. By mailing your questions to this list, your classmates will also have the opportunity to help you out. Use of this list for questions is encouraged and in some cases (e.g. midnight before an assignment is due) this may be your best choice.

[is255-ta@sims.berkeley.edu](mailto:is255-ta@sims.berkeley.edu)

Mail sent to this email list goes to the tutors and the instructor. Responses to your questions may be sent to [is255@sims.berkeley.edu](mailto:is255@sims.berkeley.edu) unless you state otherwise.

**Email the instructor or tutors directly**

Responses to your questions may be sent to [is255@sims.berkeley.edu](mailto:is255@sims.berkeley.edu) unless you state otherwise.

## 10. Grading

This course is offered on a Satisfactory/Unsatisfactory basis. To pass the course, students must complete all assignments and must attend all the lectures and labs. Points will be deducted for late assignments. If you cannot make a lecture or lab, you should email the instructor and the teaching assistants.

Class, lab, and office hour participation will be reflected in the student's grade for any given assignment and in the student's overall grade in the class. We are encouraged to see students that are actively trying to understand the materials and complete the assignments.

## 11. Testing Out of the Course

This course is required for all incoming MIMS students, unless they test out of the course or take an instructor approved substitute course (this must be an upper division course). An exam will be given to those who wish to test out of the course during the first week of class. Topics may include, but are not limited to, writing and/or debugging Java code, standard data structures, standard algorithms, binary and hexadecimal numbers, software design, and relative performance of algorithms (big-O).

## *General Information*

Anyone who has majored in computer science, who has taken a significant number of computer science classes, or has sufficient software design and development experience should test out of the class. A student should test out of the class if, for example, they could design and write a family tree program (using a tree data structure, command line input/out, use a file for saving data, support depth-first and breadth first traversal of the family tree for display or printing). Students may take the exam even if they are unsure of their abilities.

The exam will be held during the first week of class on Monday, August 30th, 2004, from 9:30 AM to 11:00 in South Hall room 205.

## **12. Java Tutors**

Students that need a tutor for the Java language should make appropriate arrangements. It is suggested that they contact another SIMS student (e.g. via the SIMS students mailing list or a U.C. Berkeley Computer Science student (e.g. Eta Kappa Nu tutoring services)).

## **13. Academic Honesty**

For many students, this will be an intensive and demanding class. At times, it may be tempting to take "short cuts" in order to complete assignments on time. Students are reminded that the instructor and the teaching assistants are here to help them! However, the students must help themselves by attempting to understand the subject matter in a timely manner, by starting assignments early so that they have enough time to get help, by taking the initiative to get help, and by getting the requisite Java programming experience before the class starts.

All students should be familiar with the Code of Student Conduct and know that the general rules stated in that document apply to this class. These rules and guidelines are available from Office of Student Life. For the purposes of assignments, it is important that students understand that they are allowed to collaborate with other students in this class; but, each student must turn in their own original work done specifically for this course (see following exceptions). In the case of team assignments, the team is allowed to collaborate with other teams (unless stated otherwise) and the team must turn in their own original work done specifically for this class. If the student or the team decides to reuse work developed by others, this must be discussed with the instructor and the reused work must be credited appropriately. For example, students have used keyboard input code from other classes rather than developing their own or using the one provided by the instructor. In some cases, the assignment may instruct to reuse work developed by others — in which case, the work should be used unless there is a compelling reason to do otherwise. If a student is found to have violated any of the academic honesty related restrictions the student may receive a failing grade on the assignment or in the course and the student's actions will be reported to

Student Judicial Affairs for administrative review.