CS18000 Home Page for Fall 2014

Problem Solving and Object-Oriented Programming

CS18000 offers an introduction to Computer Science, using the Java programming language.

Topics include primitive types and strings, selection, repetition, arrays, graphical user interfaces, methods and classes, interfaces, inheritance, exceptions, basic concurrent programming and synchronization, polymorphism, dynamic data structures, recursion and recursive data structures, and an introduction to generics.

Course Personnel

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Section</th>
<th>Instructional Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Antony Hosking</td>
<td>34295</td>
<td>Dr. Lorenzo Martino</td>
</tr>
<tr>
<td>Prof. Despoina Perouli</td>
<td>65512</td>
<td>Dr. Lorenzo Martino</td>
</tr>
</tbody>
</table>

Graduate Teaching Assistants (GTAs)

- Varun Vasudevan
- Amr Ebaid
- Sait Celebi
- Gregory Marie Essertel
- Shagufta Mehnaz
- Danushka Nirman Menikkumbura
- Mohammad-Mohsen Minaei-Bidgoli
- Michael Sen Phay
- Romila Pradhan
- Checed Aaron Rodgers
- Yongyang Yu
- Yunxiao Zou
- Chen Chiaihao
- Baharak Saberidokht
- Abdullah Khan Zehady
- Savvas Savvides
- Houzhi Xu

Supplemental Instruction

There are Supplemental Instruction (SI) study sessions available for this course. These study groups
are open to anyone enrolled in this course who would like to stay current with the course material and understand the material better. Attendance at these sessions is voluntary, but extremely beneficial for those who attend weekly. Times and locations for the study session can be found at www.purdue.edu/si or the free app www.purdue.edu/boilerguide. Students who attend these interactive sessions will find themselves working with peers as they compare notes, demonstrate and discuss pertinent problems and concepts, and share study and test-taking strategies. Students are asked to arrive with their student ID card, lecture notes and questions to these informal, peer-led study sessions.

**Grading Administrator**

Please contact TBD for the following issues:

- Web-CAT account access
- Inability to submit assignments
- Grading inconsistencies & suspected erroneous tests
- Late submission of projects if already approved by Prof. Hosking, Prof. Perouli, Dr. Martino, or your GTA

**Time/Location**

<table>
<thead>
<tr>
<th>Section</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>34295</td>
<td>MWF 0930–1120</td>
<td>EE 129</td>
</tr>
</tbody>
</table>

**Textbook**

Any up-to-date introductory textbook on Java programming. Some suggestions:

- *Java Precisely* by Peter Sestoft (second edition, 2013). ISBN 978-0262693257. This is an ideal quick reference for the reader who has learned (or is learning) Java from a standard textbook. Available through Amazon: [Paperback](#).
- *Effective Java* by Josh Bloch (second edition, 2008). ISBN 978-0321356680. This book is absolutely outstanding. It takes you well beyond the material covered in this course. However, if you're interested in doing more Java, I highly recommend that you buy this book sooner rather than later. Available through Amazon: [Paperback](#) and [Kindle](#).

**Kindle Fire**

All enrolled students are given access to a Kindle Fire for the semester. The Kindle Fire is used for selected labs and projects, and may be used for other purposes throughout the semester. Kindle Fires can be checked out from the service window near LWSN 2121.
Grades

- 40% Programming assignments (exact number may change)
  - Projects 1-3: 2% each (6% total)
  - Projects 4-5: 5% each (10% total)
  - Projects 6-8: 8% each (24% total)
- 5% Labs, Homework, in-class Clicker questions (including attendance)
  - ★ (∗) demonstrated engagement/progress/understanding through dialogue
  - ✔ (+) present
  - ✘ (−) absent
- 55% Exams
  - Exam 1: 15%
  - Exam 2: 15%
  - Exam 3: 25%

Important

Exams and labs can only be “made up” in extraordinary circumstances if arrangements are made with the CS18000 Instructional Coordinator. Such arrangements should be made at least one week before the exam. “Made up” exams must be taken before the time the rest of the class takes the same exam.

The grading for the course is statistically normalized (follows a “curve”) but the mix of grades is not fixed. If you are concerned about your grade, contact the instructor.

If you have a question about the grading of a lab or project, talk to one of your Lab Instructors about it.

Requests for regrading a lab and/or a project must be sent within two weeks from the date of the grade publication in Blackboard. The request must be sent to the Lab Instructors and cc'ed to the Instructional Coordinator (Dr. Martino).

Web-CAT

Please see this page for more information about Web-CAT, our automated grading system.

Cyberduck

Please see this page for info about using Cyberduck to download and upload files to lab machines.

i>clickers

We are using i>clickers to capture in-class quiz responses. If you don't have an i>clicker please get one ASAP. Purdue Student Government rents them or you can buy online or from the bookstores.
Please register your i>clicker online at http://www.iclicker.com.

Piazza

We are using Piazza for questions about lectures, homework, labs, projects, exams, and other class-related discussions. Sign up by visiting CS18000 on Piazza. You can post public questions visible to the entire class or private questions visible only to the instructors. You may post questions including small snippets of code (using the <code> tag), but do not post extensive pieces of code publicly.

Policies

(1) You should direct questions concerning a lab or project to a staff member, rather than a classmate. But, when you come to us, be sure that you have specific questions and can show evidence that you have spent some time on your own attempting to solve your problem.

(2) This course is worth four credit hours, including three lectures and one two-hour weekly lab. According to university guidelines, you should expect to spend, on average, an additional eight hours per week working on readings, homework assignments, and projects. Some weeks the load is less, other weeks more. Please plan your schedule carefully to avoid getting behind or missing an important deadline.

(3) ATTENDANCE IN CS18000 IS MANDATORY! You should plan on attending EVERY lecture and EVERY lab meeting. Past experience has shown us that students who attend lecture and lab regularly do better on labs, assignments, and exams - even those who think they already know the material or who think they can learn it on their own. Missed labs CANNOT be made up, unless the absence is excused. Lab absences MAY be excused for reasons of serious illness, family emergency, or official university commitments, but only if appropriate documentation is provided to one of your Lab Instructors. For planned absences (band trips, other course field trips, etc.), you must inform your instructor ahead of time, or the absence will not be excused.

(4) You should read the material in the textbook according to the class syllabus. In most cases, you will read about a concept in the book, then we will cover it in class, then you will use it in lab, then it will be used on a project, and finally it will be tested on an exam.

(5) Computers may become heavily loaded as a project deadline nears. Waiting until the last minute to work on your project is dangerous! Our CS18000 policy is NOT to extend deadlines unless most available workstations are unavailable for an extended period (like 10-12 hours) near the end of a project.

(6) Unless indicated otherwise, NO LATE projects are accepted. There are NO EXCEPTIONS to this rule except under extreme circumstances approved in advance by one of your Lab Instructors. Failure to turn in a project results in a loss of all the points allocated for the project. The same holds true for a lab assignment.

(7) In most cases no credit is given for programs that do not compile (that is, execution is suppressed due to compilation errors). Programs which execute but are not correct or complete are considered for partial credit. To receive full credit, your program must produce correct results, be well-designed, be efficient, and adhere to good programming style. Visit this link to learn about our Java
Programming Standards.

(8) WE ALWAYS WELCOME YOUR CONSTRUCTIVE COMMENTS. Please do not hesitate to bring any shortcomings to our attention.

Emergencies

In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances beyond the instructor's control. Here are ways to get information about changes in this course.

- Course web page (http://courses.cs.purdue.edu/cs18000:fall14:start)
- Instructor's e-mail (hosking@purdue.edu)
- Instructional coordinator's e-mail (lmartino@purdue.edu)

Academic Integrity

Except for team projects, all CS18000 course work must be done individually. We encourage discussion of any CS18000 topic including ideas about how to do the assignments. But, under no circumstances should you present another student's work (including program code) as your own. If you cannot do the work yourself it is unlikely that you will be successful in the exams that make up the bulk of your grade, so please seek help from the instructor or a TA if you need help with your code. Do not exchange code with any other CS18000 student via written or electronic means. Do not read another student's code in an attempt to understand your own. Superficial changes in a program (such as altering comments, changing variable names, or interchanging statements) will not escape detection by the tools we use to detect plagiarism. Even the most trivial assignment is better not done than if you cheat to complete it.

All students in CS18000 must read and electronically “sign” the Purdue University Department of Computer Science Academic Integrity Policy. This document is available at the my.cs.purdue.edu website. You need your Purdue Career Account login and password to access this page. There, after reading the policy, you indicate that you have read and understand both the policy and its consequences. There is also information about some implementation details.

Penalties

In CS18000, a first instance of academic dishonesty results in a zero for that assignment plus a letter grade deduction at the end of the semester.

A second instance of academic dishonesty results in a final grade of F.

In accordance with the Purdue University Department of Computer Science Academic Integrity Policy, all instances of academic dishonesty on an exam, project, or lab assignment are reported to the Dean of Students Office.