Kansas State University – Department of Computing and Information Sciences (CIS)
CIS 200 – Programming Fundamentals
Credit Hours: 4
Spring 2015

**Course Syllabus**

All students are expected to carefully read through and abide by all policies set forth in this syllabus. Unless there is a documented emergency (e.g. medical, funeral, etc.), no exceptions to stated policies.

Instructors: Dennis Lang
(Lectures MTW @ 8:30-9:20, 12:30-1:20)
Office: Nichols 213
Office Phone: 785-532-7768
Office Hours: Mon 9:30-11:00am

Julie Thornton
(Lecture MTW @ 1:30-2:20, 2:30-3:20)
Office: Nichols 228
Office Phone: 785-532-7785
Office Hours: Mon/Wed 3:30-4:30pm (or by appointment)

• To schedule an appointment with either instructor for questions/advising - go to schedule.cis.ksu.edu and log-in with your CIS account UN/PW. If you are a CIS major and do not yet have a CIS account, go to https://selfserv.cis.ksu.edu and simply follow the directions.

Instructors Email: dlang1@ksu.edu; juliet@ksu.edu

• Please allow at least 24-hours for a response. Replies to emails received after 4pm M-Th will typically be handled the following day. Replies to emails received after 4pm on Fri or on Sat/Sun will typically be responded to on the subsequent Monday, so please allow time for your instructor or GTA to respond.

If you have a question on the grading of a project, please initially contact the GTA for your section. If the issue is not resolved to your satisfaction, then contact your instructor. When emailing the GTA or instructor, please include your full name, the course number (CIS 200) and the lab section day/time you are enrolled. If this information is not included in your e-mail, the response will be delayed until this info can be looked-up.

Canvas Online Course Webpage: Log-in @ online.ksu.edu, click on Canvas then choose CIS 200
Lec: MTW 8:30-9:20, 12:30-1:20, 1:30-2:20, or 2:30-3:20 in Nichols 19 (see your schedule for lec time)
Lab: 2 hours on either Wed, Thur, or Fri in Nichols 21 or N122 (see your schedule for lab time/location)
Exams: Three course exams will be given this semester during Weeks 4, 8, & 13 in DUF 1107 (Fiedler Aud) @ 7:00pm (see course schedule at the end of the syllabus for exact dates)

Prerequisite: MATH 100 (College Algebra) and either CIS 115 (Intro to CS) OR ECE 241 (Intro to Computer Engineering). These are prerequisites that are considered essential to student success in the course (CIS 111 is also acceptable). If you have not taken and earned a ‘C’ or better in (or received transfer credit for) the prerequisite courses, you should only be in this class with permission from the instructor. Otherwise, I strongly suggest you consider taking CIS 115, ECE 241 or CIS111 (see below) first. Students are expected to have the logical thinking skills that come from applying techniques to solve mathematical problems. Logical thinking skills will be used when writing computer programs.

Important: This is a 200-level course intended as the first programming course for Computer Science, Information System, and Engineering majors. Non-majors interested in programming should consider CIS111 Introduction to Computer Programming, a 3 credit 100-level course that moves at a much slower pace than CIS200 and does not cover as much material (counts as an unrestricted elective for CIS majors). Compare to taking Math 205 (Gen Calculus) versus Math 220 (Calc I).
**Objectives:** Learn the fundamentals of writing computer programs primarily using the Java programming language (C# is briefly introduced at the end). The skills developed in this course are transferable to subsequent programming language courses. Specifically, the following topics will be covered:

**Weeks 1-4:** Command Line Compiling vs. using an IDE; Introduction to Java (JVM); Variables and operations; Output / User Input; Conditional statements; Loops

**Weeks 5-8:** Arrays; Strings; Reading from and writing to text files; Pre-built and Programmer-defined Methods; Command Line Arguments; Variable Length Parameters

**Weeks 9-15:** Classes and Objects; Program design; Exception Handling; Collection Classes (ArrayLists, TreeSet, and HashMaps); Recursion; C# vs. Java / Introduction to Visual Studio

**TEXTBOOK AND MATERIALS:**
In addition to the texts listed below, there is also a “Java Reference” folder available within the ‘Files’ folder in Canvas, written and developed by Julie Thornton previously for this course.

There are two **required** textbooks for this course:

   - *This text will often be referenced in lecture, so good to bring to every class!*

2) “Programming with Greenfoot” customized textbook*

Both the campus union bookstore (www.bkstr.com/) and Varney’s (varneys.verbacompare.com/) carry the required texts for this course. If you can find a new or used copy of the same book elsewhere, that’s fine! Several sites like Course Smart (coursesmart.com) and Amazon allow you to rent a digital version of the text (eText) at less than purchasing the physical text (Search by ISBN). eTexts gives you access to your textbook on laptops, tablets and mobile devices.

*Programming with Greenfoot* will be used in lab ONLY. A customized (cheaper) version of the text created for this course is available through the K-State Union bookstore or Varney’s. You can get a special price if you purchase both the physical and e-text from the bookstore. If you can find a cheap used copy of Michael Kolling’s *Intro to Programming with Greenfoot* (ISBN: 0136037534) it is the same book with more chapters. Page numbers will vary slightly from the customized text.

Besides the required Gaddis textbook, two other good Java texts I recommend as a good reference are:


*Note: Older editions of these texts are fine and can be found for almost nothing online!*

**CodeLab:** You will be using an online learning tool called CodeLab for some of your homework problems. The cost to access CodeLab for the semester is $25 *(if purchased online)* or $27 *(if paying by check)*. Details on how to register and use CodeLab is discussed in a separate document posted in the Files folder in Canvas.

**Programming Language/Software:** Initial projects will be written in Java and submitted as zip files. All programs must compile and run from the command-line using the *javac* command. If you are using Windows, you will be able to compile programs using the command-line compiler. You are also welcome to use Eclipse IDE *(Integrated Development Environment)* to develop programs (free and available for Windows, Linux or Mac). *Dr. Java* and *BlueJ* are also free. We will also be using a free software product called Greenfoot (free download at Greenfoot.org).
The final project will be written in C# and submitted as a zip file. It must compile and run from the command-line (using csc in Windows and Mono with a Mac). You are welcome to use Visual C# Express (Windows) or MonoDevelop (Mac) to help write the final project. Both are free. More information on writing C# programs will be made available towards the end of the semester.

More information on compiling and running Java programs is available in the Files folder in Canvas. More information on writing C# programs will be made available towards the end of the semester.

Grading Info

Incompletes given only with documented extenuating circumstances (such as a medical emergency) at the discretion of the instructor. You must explain and document why dropping the course before the final drop date was not an option. (Incompletes are NOT given because you do not want an ‘F’ on your transcript. Be aware that although the ‘F’ grade remains on the transcript, a course can be retaken and the last grade earned in the course will be used in GPA calculation, replacing the original grade.) If desired, it is your responsibility to drop the course. No “automatic” drops due to non-attendance. No incompletes given after the final day of class.

The gradebook in Canvas is updated on a regular basis. It is expected that each student will frequently monitor their points for accuracy throughout the semester and bring any discrepancies to the instructor attention immediately. Please do not wait until the end of the semester (or after the course ends) to mention a discrepancy.

Grades are based upon the following weights:

- Lab activities (11-drop lowest)..... (15%)
- Programming Projects (10)........ (15%)
- Codelab (200/250 required)....... (10%)
- Course Exams (3).................... (40%) (i.e.13.3% each)
- Comprehensive Final exam (1).... (20%)

Important: Exams are the only assessment component of the course that is completed under supervision without any outside assistance. Therefore, exams are used as a primary measurement in your understanding of the course material and in determining your final grade. Please be aware that 60% of your final grade is determined by how well you do on the course exams, which require you to write code by hand (no compiler). If you do poorly on the exams, it increases your chances of not get the minimum grade of ‘C’ in the course required to continue on to CIS 300. Lab activities, Programming Projects, and Codelab (40%) are primarily learning tools used to help you better prepare for the exams. If you attend all lectures, successfully do all of the Codelab exercises, and complete and understand all weekly labs and programming projects, you should do well on all exams.

Grading Scale:

- A: 90% or above
- B: 80%-89.9%
- C: 70%-79.9%
- D: 60%-69.9%
- F: Below 60%

**For CIS majors, be aware that a ‘C’ or better is required in this course to continue on into CIS 300**
Description of COURSE WORK

**CodeLab:** There are 250 total exercises on CodeLab. While I strongly recommend that you do ALL the exercises through-out the semester, you must correctly solve at least **200** of the 250 exercises to get 100% of the Codelab points. Codelab is a learning tool not a testing tool. You can attempt to answer each exercise as many times as needed until you get it right. You can seek assistance if you are having trouble with any exercise. To check your progress in CodeLab, choose Help–Account Info–Exercise Summary.

Due dates for Codelab exercises are listed on both the course schedule AND within CodeLab and will NOT be accepted after 11:59pm on that date. No exceptions since solutions will be made available the following day. Due dates are in place to encourage students to regularly do the exercises that correspond to what is being done in lecture to help better prepare you for your exams. Feel free to work ahead, if desired. **More info on CodeLab is posted in the files folder in Canvas.**

**Note:** Students who took the class previously must purchase a new Codelab access code and redo all assigned exercises to help them properly prepare for exams.

**Lab Activities:** There will be 11 lab activities throughout the semester led by a Graduate Teaching Assistant (GTA). **Attendance in Lab is required for credit and labs may not be made up.** You have to be present to learn what is taught for that lab period. However, to allow for an unforeseen emergency, sickness, conflicts, etc. your lowest lab score will be dropped from the final grade calculation.

**Labs should take a minimum of one hour to complete** – therefore, **if you do not attend at least an hour of lab, a minimum of 3 pts (15%) will be deducted from your final score:** likewise, **if you cannot complete the lab in the allotted 2 hour period, a minimum of 3 pts (15%) will be deducted from your final score.** The final score you received on a lab is left to the discretion of the GTA of your section.

**IMPORTANT!** Before attending lab **Week 1**, you will need a CIS account (if you don’t have one). Go to [https://selfserv.cis.ksu.edu/](https://selfserv.cis.ksu.edu/) to request that an account be created. If you forgot your CIS account password or get locked out, you can now go to [https://password.cis.ksu.edu/](https://password.cis.ksu.edu/) to resolve the problem yourself. Any other help requests should be sent to support@cis.ksu.edu or help@cis.ksu.edu. System administrators are located in Nichols 116-119 if you have problems or questions.

**Programming Projects:** There will be **10** required, **individual** programming projects throughout the semester plus **one** extra credit project. Project descriptions will be posted on Canvas and will be discussed during class. All projects must be submitted as a compressed file (.zip) within Canvas. Be aware that a score of zero may result if submission guidelines are not followed and projects are not properly submitted. While this may seem harsh to some students, the goal is to set a precedence for subsequent course requirements and get students to focus on properly submitting their work as requested by their instructor.

**Late projects will be accepted but penalized 10% for each day it is late (up to 30%) and will not be accepted after 3 days from original due date.** A project submitted after 11:59pm (CST) on the date a project is due is considered late. All students are expected to abide by this policy, with no exceptions. **Be aware that personal computer/internet problems are NOT a reason for missing a project deadline.** Allow yourself enough time to address problems (if they arise) by **starting all programming projects as soon as possible and seeking help early in the week if you have questions.** If you procrastinate and wait until the day a project is due to begin working on the assignment, issues may arise which causes you to submit work late (or not at all). It is very important to learn to work with deadlines as you will face them continuously in subsequent coursework as well as in business and industry.

**ALL projects are expected to be done individually – they are not team projects.** Working with another student may result in a violation of academic policy. **I warn against EVER showing your code to any student.** Please refer to the Academic Integrity Policy within this syllabus which states, “If two (or more) students are involved in ANY violation of this policy, **ALL** students involved receive a **zero** for the assignment with an official warning report sent to the K-State Honor Council. The second offense results in a failing grade for the course.”
**Exams:** There will be three course exams and one comprehensive final exam given throughout the semester. All exams in this course and subsequent programming courses are hand-written and done without the use of a computer (i.e. no compiler), so properly prepare for these exams! Exams are closed-book/note unless otherwise announced. For the three course exams, all students are required to take the exam on Wed. evening from 7:00-8:50pm in Fiedler 1107. Please contact your instructor at least 72-hours in advance if you have a conflict with an exam time. Lecture will not meet on exam day and labs will not meet on exam week.

Makeup exams will only be permitted with prior authorization from the instructor AND documented extenuating circumstances. Please be aware that a different exam will be given as a make-up.

---

**Course Policies**

**Lecture Policy:** **Please be respectful to the other students and your instructor by refraining from talking during lecture. (It's only 50 minutes!)** This includes talking about the class, the lecture, an assignment, etc. – please wait until after the class is over to discuss. Students that talk during lecture distract both the instructor and those students around them, as well as miss out on what is currently being discussed. If you have questions, please ask ME rather than the student next to you. Questions are an excellent learning tool for all. I will remind students that talk during lecture to refrain from talking. If you continuously violate this policy, you may be asked to not attend the lecture. I encourage you to bring your laptop to take notes and enter the code as it is created. However, if you choose not to pay attention to the lecture, that is your choice, but please do not distract those around you from learning.

**Attendance Policy:**

“We have a contract here...you expect to be taught and...I agree to teach you. When you are not present, our contract is in breach. You cannot learn and I cannot instruct. Your presence in this class is, therefore, essential.” (~From the 1970’s TV Series “The Paper Chase”)

You are responsible for all material presented in lecture and lab. If you must (or chose) to be absent, please contact a fellow student to see what was covered. Announcements and code created in class are posted after lecture daily on Canvas. It is the student's responsibility to come to class and actively take notes to supplement any code discussed in class.

**Expectations for Classroom Conduct:**

All student activities in the University, including this course, are governed by the Student Judicial Conduct Code as outlined in the Student Governing Association By Laws, Article VI, Section 3, number 2. Students who engage in behavior that disrupts the learning environment may be asked to leave the class.

**Accommodations:**

Students with disabilities who need classroom accommodations, access to technology, or information about emergency building/campus evacuation processes should contact the Student Access Center and their instructor. Services are available to students with a wide range of disabilities including, but not limited to, physical disabilities, medical conditions, learning disabilities, attention deficit disorder, depression, and anxiety. If you are a student enrolled in campus/online courses through the Manhattan or Olathe campuses, contact the Student Access Center at accesscenter@k-state.edu (785-532-6441); for Salina campus, contact the Academic and Career Advising Center at acac@k-state.edu (785-826-2649).

**Campus Safety Statement:**

Kansas State University is committed to providing a safe teaching and learning environment for student and faculty members. In order to enhance your safety in the unlikely case of a campus emergency make sure that you know where and how to quickly exit your classroom and how to follow any emergency directives. To view additional campus emergency information, go to the University's main page (www.ksu.edu) and click on the Emergency Information button.
Academic Honesty / Academic Integrity Policy

“[Plagiarism] is theft, plain and simple … It is theft of another writer’s ideas, work, and time; it is theft of [your] fellow students’ time; it is theft of [your] own time, honor, and education; and it is a theft of your [professor’s] time.” - Segal, C. (Sept. 15, 2006). Chronicle of Higher Education; B5

Academic honesty and integrity are essential to the existence and growth of the academic community. Academic dishonesty violates the most fundamental values of an intellectual community and depreciates the achievements of all graduates from that institution.

Kansas State University has an Honor System based on personal integrity, which is presumed to be sufficient assurance that, in academic matters, one's work is performed honestly and without unauthorized assistance. Undergraduate and graduate students, by registration, acknowledge the jurisdiction of the Honor System. The policies and procedures of the Honor System apply to all full and part-time students enrolled in undergraduate and graduate courses on-campus, off-campus, and via distance learning. The honor system website can be reached via www.ksu.edu/honor. A component vital to the Honor System is the inclusion of the Honor Pledge which applies to all assignments, examinations, or other course work undertaken by students. The Honor Pledge is implied, whether or not it is stated: "On my honor, as a student, I have neither given nor received unauthorized aid on this academic work." A grade of XF can result from a breach of academic honesty. The F indicates failure in the course; the X indicates the reason is an Honor Pledge violation.

The only way to truly learn the concepts for this course is to do ALL of your own work, especially your own programming and debugging/problem solving. You are not only learning Java, but developing problem solving skills and program development skills. All assignments in this class are expected to be individual work, unless clearly stated otherwise. They are NOT team projects. Helping a friend complete his/her project can quickly cross the line of unauthorized collaboration, so be very careful on the assistance you offer. Don’t share your code with a friend! Help sessions and office hours are available for those students requiring extensive help – it is not your responsibility to help a friend pass the course!

Using the internet to help answer questions that arise is fine, since it is frequently done in industry. However, claiming any work that is not your own is plagiarism and is a violation of the K-State Honor Code. If you find and use an algorithm or code on an external website, it is allowed as long as you accurately cite the website URL within a comment above the code or algorithm. Failure to do so will be considered plagiarism and will be considered a violation of the Honor Code. (Obviously, friends, roommates, a spouse, significant other, etc. do not count as a reference and will be considered unauthorized collaboration).

All students are expected to abide by this policy. My goal is to ensure that honest students are not disadvantaged by students who violate this policy. Assistance from a tutor, student or any other source must not cross the line of unauthorized collaboration (usually defined as “Any act of two or more students actively cooperating on any assignment when such cooperation has not been expressly permitted by the instructor”) plagiarism (the act of representing someone else's work as one's own) or copying, which includes any of the following:

1. Any act of copying info from another student by any means to obtain an advantage for one's self.
2. Any act of conveying info to another student that provides an unfair advantage to that student.
3. Any act of representing others' work, whether copyrighted or not, as one's own.

Be forewarned: If two (or more) students involved in ANY violation of this policy, ALL students involved receive a zero for the assignment and the offense is officially reported to the K-State Honor Council. At the discretion of the instructor, an XF in the course may also be given. The second offense categorically results in a failing grade for the course (XF) and possible suspension from the university (this decision is made by the K-State Honor Council).
My Advice: **If it is not your work, don’t submit it!** Points gained are not worth the penalty that ALL parties incur if caught. Thoroughly think through the consequences if you were to get caught. Students who do not work through the process of doing their assignments individually rarely do well on the exams and, if they still manage to squeak by with a ‘C’ in this course, they struggle (and often fail) subsequent coursework. Even if you are fortunate enough to not caught violating policy, you are only hurting yourself in the long run.

*The ethical path (choice) is rarely the easy path (choice). Go beyond legality (avoiding punishment) and do what is right. There are no shortcuts in learning. You must put in the work.*

---

**Computer Availability and Support**

1) Before seeking assistance, make sure you have followed all instructions in the materials posted within Canvas, such as the *Compile Guide, using Codelab, etc.*

2) The CIS department, the course instructor, and the Graduate Teaching Assistants are NOT responsible for technical issues related to students’ personal computers. If you need help with your personal computers, the IT Help Desk is located in 214 Hale (785-532-7722).

3) **Personal computer/internet problems are NOT a valid reason for missing an assignment deadline.** If you experience problems with your personal computer, use any of the public computing labs on campus. For info on additional labs on campus, visit: [http://www.k-state.edu/its/labs](http://www.k-state.edu/its/labs).

<table>
<thead>
<tr>
<th>Lab locations</th>
<th>Hours</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nichols Hall, Room 21</td>
<td>24/7 (except when reserved for class)</td>
<td>PCs &amp; Unix</td>
</tr>
<tr>
<td>Fiedler Engineering Library</td>
<td>See <a href="http://www.lib.k-state.edu/fiedler-engineering-library">http://www.lib.k-state.edu/fiedler-engineering-library</a> for hours</td>
<td>PCs</td>
</tr>
</tbody>
</table>

---

**Final Thoughts...**

*Procrastination/Poor time management on your part will NOT constitute special consideration on my part*

All students are expected to follow the *same* due dates and deadlines. It is not fair to grant one student an extension or special consideration without offering the same opportunity to all students. Poor time management is a major stumbling block for many students. Do not procrastinate and wait until the night before an assignment is due to begin the assignment. By allocating yourself only a short amount of time to complete an assignment, additional pressure is added, lessening the learning opportunity. Always keep in mind: *It's your education…you get out of it what you put into it.*

*Seek knowledge first ... a good grade often follows*

*Truly* put your best effort towards learning the information, seeking understanding and knowledge first. A good grade is usually the by-product of such effort.

*If you really want (or need) to do something, you find a way; if you don’t, you find an excuse*

Life is all about choices and *all choices have consequences*, both good and bad. For some, computer science is your major. Be aware that the courses do not get any easier as you progress through your academic career so take as much as you can from this class while you are taking it. Make wise choices *all semester* and don’t get overwhelmed by what one new student described as the “magic of college”

*Worry about your grade ALL SEMESTER not just the last few weeks of class*

Your grade is a *semester long* achievement. *Incompletes* are not given. If you lapse at the beginning, middle, or end of the course, your grade *will* suffer. You can’t make up un-submitted projects, make up un-attended labs, or re-take failed exams as the semester nears the end. While extra credit is offered through-out the semester, there is no ‘extra’ extra credit. Nothing can be done after the final exam is taken to raise a 69% to a ‘C’. *Worry about your grade ALL SEMESTER, not just the last few weeks of class.*
<table>
<thead>
<tr>
<th>Wk</th>
<th>Date</th>
<th>Topics Covered/Activity</th>
<th>CodeLab Topics (250) / Due Date Deadlines</th>
<th>Labs (Topics)</th>
<th>Project Deadlines (Subject to Change)</th>
</tr>
</thead>
</table>
| 1  | 1/20 | Intro to Java / JVM Fundamentals / Console I/O  
Reading: Chapters 1 & 2 | Reminder-200/250 Codelab exercises must be completed during the semester to get 100% of the points | Lab 1 (Console IO) | Lab 1 (Console I/O)                           |
| 2  | 1/26 | Finish Fundamentals / Console I/O Program Development  
Decision Structures  
Reading: Chapter 3 | 1) Codelab Warmup (4)  
2) Variables & Types (16)  
By 11:59pm, Wed Jan 28th | Lab 2 | Proj 1: Fri-Jan 30th (Eclipse) |
| 3  | 2/2  | Decision Structures / GUI I/O Loops / Nested Loops  
Reading: Chapter 4 (Sec 4.1-4.9) | 3) Input and Output (13)  
4) Operations (48)  
By 11:59pm, Wed Feb 4th | Lab 3 (ifs/Loops GUI I/O) | Proj 2: Fri-Feb 6th (Conditionals) |
| 4  | 2/9  | Finish Loops / Exam Review  
** EXAM 1: Wed, Feb 11th @ 7pm in Fiedler Aud ** | 5) Conditionals (14)  
6) Loops (14)  
By 11:59pm, Tue Feb 10th | No Lab | Proj 3: Fri-Feb 13th (Loops) |
| 5  | 2/16 | Arrays / Strings  
Reading: Chap 7 (7.1,7.2,7.4,7.6) | 7) Arrays (37)  
By 11:59pm, Mon Feb 23rd | Lab 4 | Lab 4 (Arrays) |
| 6  | 2/23 | File I/O / Pre-built Methods  
Reading: Chapter 4 (Sec 4.10)  
Feb 24: Last day to drop w/o a ‘W’ | 8) Strings (12)  
By 11:59pm, Mon Mar 2nd | Lab 5 (File I/O) | Proj 4: Fri-Feb 27th (Arrays, Strings, Text File I/O) |
| 7  | 3/2  | Programmer-defined Methods / Sorting algorithms  
Reading: Chap 5, Chap 7 (7.3, 7.11) | 9) Static Methods (32)  
By 11:59pm, Tue Mar 10th | No Lab | Proj 5: Fri-Mar 13th (Methods) |
| 8  | 3/9  | Cmd-Line args/Var Len Parameters  
Reading: Chapter 7 (Sec 7.12)  
Exam Review  
** Exam 2: Wed, Mar 11th @ 7pm in Fiedler Aud ** | ** Greenfoot text required in lab this week!** | No Lab | Proj 6: WED-Apr 1st (Greenfoot) |
| 9  | 3/16 | Classes and Objects  
Reading: Chapter 6 (Sec 6.1-6.7)  
Introduction to Greenfoot | ** Greenfoot text required in lab this week!** | No Lab | Proj 7: Fri-Apr 10th (Classes / Objects) *Available Week 10* |
| 10 | 3/30 |Classes and Objects (cont.)  
Reading: Chapter 7 (Sec 7.7)  
Chapter 8 (Sec 8.1-8.6) | Mon, Mar 30th: Last day to drop a 16-week course | Lab 8 (Classes & Objects) | Proj 8: Fri –Apr 17th (Greenfoot) (Extra Credit Project Available Week 12) |
| 11 | 4/6  | Program design (Model-View-Controller) / Exception Handling  
Reading: Chap 11 (Sec 11.1-11.2) | 10) Classes & Objects (25)  
By 11:59pm, Mon Apr 6th | Lab 9 (MVC) | Proj 9: Fri –Apr 24th (Exceptions / ArrayLists) |
| 12 | 4/13 | Exception Handling (cont.)  
Collection classes – ArrayLists, TreeSet, HashMaps  
Reading: Chapter 7 (Sec 7.13) | 11) Program Design (27)  
By 11:59pm, Mon Apr 13th | Lab 10 (Exception/ArrayLists) | Proj 10: Wed-May 6th (C#) |
| 13 | 4/20 | Collection classes / Exam Review  
** EXAM 3: Wed, Apr 22nd @ 7pm in Fiedler Aud ** | 12) Exceptions (8)  
(Last One!)  
By 11:59pm, Tue Apr 21st | No Lab | Extra Credit Project: Mon-Apr 27th (Greenfoot) |
| 14 | 4/27 | C# vs. Java | ---- | Lab 11 (C#) | Final exam: Thurs., Dec 18th 7:30-9:20am (Location to be announced in class) |
| 15 | 5/4  | Recursion (Reading: Chapter 15)  
Final Exam Review | ---- | No Lab | Proj 10: Wed-May 6th (C#) |
| 16 | 5/14 | | | | |