CSCI335 User Interface Design & Programming (Fall 2014)

Contents
1. Professor/Class Information
   1. Class Information
   2. Professor Information
   3. Departmental Information
   4. Relevant Websites
2. Course Description
3. Course Policies
4. Textbooks and Resources
   1. Required Textbooks
   2. Supplementary Textbooks
      1. Android Testing
      2. Interface Design
   3. On the Bundle of Services Once Provided by a College Bookstore
   4. Other Resources
5. Learning Tasks
   1. Task 01. Class Attendance and Participation
   2. Task 02: Weekly Reading Assignments
   3. Task 03: Biweekly Programming Assignments
      1. Task 03A: Build the Swing Examples
      2. Task 03B: Busy Box
      3. Task 03C: Layouts
      4. Task 03D: GUI Program Design
   4. Task 04: Mid-Term User Interface Design Project
   5. Task 05: Swing Application
   6. Task 06. Exams
      1. Task 06A: Mid-Term Exam
      2. Task 06B: Final Exam
   7. Task 07. Special Project
6. Learning Outcomes
   1. Learning Themes in This Course
   2. Course Alignment with Program Learning Outcomes
   3. Course Objectives and Student Learning Outcomes
      1. IDEA Objective #1: Gaining factual knowledge (terminology, classifications, methods, trends).
      2. IDEA Objective #2: Learning fundamental principles, generalizations, or theories.
      3. IDEA Objective #3: Learning to apply course material (to improve thinking, problem solving, and decisions).
      4. IDEA Objective #4: Developing specific skills, competencies, and points of view needed by professionals in the field most closely related to this course (that is, user interface design and implementation).
7. Grading
   1. Standard Rubrics
      1. Rubric for Instructor Assessment of Biweekly Programming Assignments
      2. Rubric for Instructor Assessment of Oral Communication
   2. Computation of Final Grade
      1. Computation of Maximum Possible Points
8. Course Schedule & Due Dates

1. Professor/Class Information

1.1. Class Information

Professor: Bill McCarty, Ph.D.
Title: User Interface Design Programming
Credit Hours/Units: 3
Term: Fall 2014
Course Code: CSCI335-01
Classroom Location: Library PC Lab, LIBRARY 141
1.2. Professor Information

| Office Location: | McNally, White 48C |
| Office Hours: | TR 3 p.m. - 4:15 p.m. & by arrangement |
| Office Phone: | 562-903-4741 |
| E-mail: | william.mccarty+csci335@biola.edu |

1.3. Departmental Information

| Teaching Ass't: | None |
| Admin. Ass't.: | Jerri Smith |
| Phone: | 562-903-4741 |
| E-mail: | jerrianne.smith@biola.edu |
| Lead Alcove TA: | Kellian Hunt |

1.4. Relevant Websites

| University Website: | http://www.biola.edu |
| Math/CS Dep't. Website: | http://csci.biola.edu |
| Prof. McCarty's Website: | https://compsci-wm1.biola.edu/moin/ |
| Website for this course: | https://compsci-wm1.biola.edu/moin/CSCI335 |
| User ID: | biola |
| Password: | grace2u |
| Canvas LMS: | https://canvas.biola.edu/ |

2. Course Description

CSCI335 User Interface Design & Programming. User interface design, implementation, and evaluation; event-driven programming in GUI applications and web applications; user-centered design methodologies.

Prerequisite: CSCI106 Data Structures (or permission of instructor)

Offered: Alternate years.

Note, however, that the course has recently been offered annually, in the fall. I suppose this will continue so long as enrollments support this more frequent offering.

3. Course Policies

Because the course policies are common to several courses, they reside in a single, separate document (an application of the "Database Principle," which you will encounter in this course). See Prof. McCarty: Course Policies, Fall 2014

4. Textbooks and Resources

4.1. Required Textbooks

  - ISBN-10: 1118766571
  - Amazon URL: http://www.amazon.com/dp/1118766571
This book was just released, on September 2. Do not conflate it with the 3rd edition
By the way, have you learned the syntax for the Amazon URL of a book?
○ Safari Books Online does not yet have the 4th ed. If you're using Safari, we'll work out a reading list for the 3rd ed.
○ Note that that 4th ed. adds material on mobile devices.

  ○ ISBN-10: 1449319238
    ○ Pbk.: $23. Too new to be readily available used at a good price. Kindle ed.: $16; rents for about 50% of this cost. Be sure to get the 2014 edition.

  ○ ISBN-10: 1449316646
    ○ Pbk.: $34; rent: $ 14. Used copies readily available. Kindle ed.: $20; rents for about 50% of this cost.
  ○ From the Preface: "The purpose of this book is to enable you to create well-engineered Android applications that go beyond the scope of small example applications." And a few pages later: "By the time you reach the end of this book, we want you to have gained knowledge beyond reference material and a walk-through of examples. We want you to have a point of view on how to make great Android applications." And yet again near the end of the first chapter: "Some examples you will read fall short of what you will need for creating the best possible extensible and maintainable commercial software. Many example applications make choices that make sense if the coder's goal is to create an example in a single Java class. In many cases, Android applications are overgrown versions of example code [emphasis added], and they end up unreadable and unmaintainable. But that does not mean you should avoid reading examples that are more expedient than a large application should be." Such wisdom should help you see why I've assigned this book.
  ○ Electronic supplements: [https://github.com/bmeike/ProgrammingAndroid2Examples](https://github.com/bmeike/ProgrammingAndroid2Examples).

4.2. Supplementary Textbooks

4.2.1. Android Testing


4.2.2. Interface Design

  ○ Kindle ed.: $28; readily available used.
  ○ Describes a design process based largely on Cooper, Reimann, & Cronin (2007) that focuses on simple explanation and practical advice.
  ○ As discussed in class, accessibility—making user interfaces usable by those having one or more disabilities—is a major concern for governmental and educational systems; it is increasingly a major concern for other system types, as well. Neglect accessibility at your peril.
  ○ A fun-to-read classic. Widely and cheaply available used.
  ○ [https://docs.google.com/file/d/0Bx1Dajl-4OWQdHBWTWVnTHlKMDg/](https://docs.google.com/file/d/0Bx1Dajl-4OWQdHBWTWVnTHlKMDg/).
  ○ Note: ACCESSIBLE ONLY TO BIOLA.EDU USERS. This is a self-published HTML edition of the now out of
The following chapters are included in the downloadable file, above. They are called to your attention for possible value but are not associated with any reading assignment.

  
  Note: If you haven't programmed in Java, you may find this a useful, quick summary of Java.

  
  Note: This chapter provides a quick, alternative perspective on the user interface design process.

  
  Usability testing is the way to evaluate and troubleshoot a user interface design. Unlike many authors, who offer $1 million usability test plans, Krug offers a practical, usability-testing-in-the-small approach.

  
  Includes content on mobile as well as web design, despite the title. Revised edition of a classic. Widely available used.

  
  The book that taught me user interface design. Brilliant in its elegant simplicity. An example: "Reduction through successive refinement is the only path to simplicity." Focused on static, visual aspects of design. Widely available used.

  
  For Android design and programming you simply must have this book. However, I will share selected highlights of the book with you during the term. I suggest you wait to buy the book at such time as you begin developing mobile software apart from class assignments.
  
  During presentations, students should follow along with their own phones and actively enhance the activity by means of clarifying demonstrations.

  My private PDF copy: [https://drive.google.com/a/hagiasmon.org/file/d/0ByqembXsPPzOWXZkZFRwT3gwVUU/edit?usp=sharing](https://drive.google.com/a/hagiasmon.org/file/d/0ByqembXsPPzOWXZkZFRwT3gwVUU/edit?usp=sharing).

  
  A classic that can change the way you think about design generally, not only user interface design. Freely available, so there's no excuse to omit this one.


  Interface design for Android is different, in important respects, from interface design for the desktop and web. This book focuses on those differences and how to deal with them;


  - Note: I have diligently attempted to locate a less expensive substitute textbook but ultimately failed to do so because the comprehensive scope of this book is unparalleled. As far as I know, this book is not legitimately available as an ebook except on a rental basis. Amazon is one source for a rental copy.
  - The "practitioner's summmary" of chapter 1: If you are designing an interactive system, thorough user and task analyses can provide the information for a proper functional design. A positive outcome is more likely if you pay attention to reliability, availability, security, integrity, standardization, portability, integration, and the administrative issues of schedules and budgets. As design alternatives are proposed, they can be evaluated for their role in providing short learning times, rapid task performance, low error rates, ease of retention, and high user satisfaction. Designs that accommodate the needs of children, older adults, and users with disabilities can improve the quality for all users. As your design is refined and implemented, evaluation by pilot studies, expert reviews, usability tests, user observations, and acceptance tests can accelerate improvement. Success in product development is increasingly being measured in terms of hard evidence that universal usability is being attained, (rather than testimonials from a few enthusiastic users). The rapidly proliferating literature and sets of evidence-based guidelines may be of assistance in designing your project while accommodating the increasingly diverse and growing community of users."
4.3. On the Bundle of Services Once Provided by a College Bookstore

Formerly, the adopted version of each course textbook was listed for sale in the campus bookstore. But, because bookstore prices were not competitive and/or for other reasons, it came to pass that only about 5% of my students actually purchased their textbooks via the bookstore. The bookstore must return unsold textbooks at its own cost. Therefore, unsold textbooks put pressure on tuition costs. I have responded by discontinuing the practice of listing books via the bookstore. Nevertheless, I expect that you will timely obtain copies of the necessary course materials and by "timely" I mean by the next class session. Whether or not this can be achieved when books are ordered online, sometimes with quoted delivery times in excess of two weeks, cannot be my concern. I'm not given leeway to truncate course objectives to accommodate purchase preferences. So, I respectfully suggest that you spend your textbook dollars with a view to reliable, timely delivery.

I do understand, I think, why you choose to purchase books off campus. In your place, I'd almost certainly do the same. But certain consequences--some of them undesirable--flow from your decision. I'm not in a position to insulate you from those effects without causing even more serious problems. So you're going to have to bear, if you will, the weight of your decision. You need to somehow purchase the right book. And you need somehow to get it here yesterday.

4.4. Other Resources

- **Personal Computer (PC).** To complete the in-class and homework assignments for this course, you will need a personal computer (PC) manufactured within the last several years and capable of running virtualization software, such as Oracle's VirtualBox. See [https://www.virtualbox.org/](https://www.virtualbox.org/). VirtualBox is capable of running under recent versions of Microsoft Windows, Apple OS X, and major Linux distributions (e.g., Red Hat, CentOS, Fedora, Debian, and Ubuntu). Frankly, I find VMware to be a more reliable virtualization platform. But in deference to those mistaken souls who own the least suitable operating system for a computer science student, Mac OS X, I will prepare and distribute VirtualBox virtual machines and will make reference to VirtualBox in class-related demonstrations and activities using virtualization. These are many in this course because we want the operating system, the subject matter of the course, to be as visible and pliable as possible--something Microsoft abhors and Apple manages largely to prohibit. In certain practical respects, your knowledge of system administration, in particular, consists in the number of ways that you have broken systems and subsequently recovered them. Virtualization expedites the breaking and, to a lesser degree, the fixing or recovery of systems. So in this course we have a considerable appetite for virtualization.

- **Support for Virtualization.** Virtualization also affords us the opportunity to standardize our development environments and toolchains, so that we can be on the same page, so to speak. As far as I know, every desktop and notebook (not necessarily netbook or nettop and most probably not tablet) computer manufactured in the last several years has support at the hardware level for virtualization. Popular desktop and notebook operating systems (Windows Vista and later, OS X, and recent releases of mainstream Linux distributions such as Red Hat, CentOS, Fedora, Debian, and Ubuntu) support the virtualization programs we will be using.

- **Biola-Provided Resources.** Note that instructional classrooms are available to students whenever a class is not in session. Also a small lab of computers, in the "Alcove," is available to you 24x7. See Kellian Hunt, who is the lead Alcove TA (SysAdmin) this term.

- In addition, the Biola Library’s website provides access to thousands of electronic books and journal articles for your research. The library’s home page is available at: [http://library.biola.edu](http://library.biola.edu).

5. Learning Tasks

5.1. Task 01. Class Attendance and Participation

Class Attendance and Participation do not affect the course grade unless or until they're unsatisfactory, in which case they can result in a failing grade. For more information about expectations with respect to Class Attendance and Participation, see Course Policies for Fall 2014.

5.2. Task 02: Weekly Reading Assignments

The following table identifies the course reading assignments, which are due as indicated on the Course Schedule, below.

<table>
<thead>
<tr>
<th>ID</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>Shneiderman: 1</td>
</tr>
<tr>
<td>R02</td>
<td>Cooper: 1-3; Gargenta: 1-4</td>
</tr>
<tr>
<td>R03</td>
<td>Cooper: 4-6; Gargenta: 5-7</td>
</tr>
<tr>
<td>R04</td>
<td>Cooper: 7-9; Gargenta: 8-12</td>
</tr>
<tr>
<td>R05</td>
<td>Cooper: 10-11; Gargenta: 13-15</td>
</tr>
<tr>
<td>R06</td>
<td>Cooper: 12-13; Gargenta: 16; Mednieks: 1-5</td>
</tr>
<tr>
<td>R07</td>
<td>Cooper: 14-15; Mednieks: 6-9</td>
</tr>
</tbody>
</table>
Note: To "study" means to do those things necessary to facilitate understanding and long-term retention of at least the contours of the content read, with a view to application of the content (that is, to changed behavior). Because learning styles and preferences differ from person to person the things necessary likewise differ. I am trying to hold you accountable for the results rather than the means of learning.

Also submit:

- Your answer(s) to any related discussion question(s).
- Your study notes for the chapter. These may take any form you prefer and so the term "notes" may not be fully apt. However, I strongly recommend the use of a simple outline.

Assessment: The grading of reading assignments will be based mainly on your own evaluation of your work. However, I reserve the right to give or withhold credit based on my own evaluation of your work. If I deem your work to be unsatisfactory, I may permit you, at my sole discretion, to re-submit the work. My decision will be based on my assessment of the quality of your work and extent of your effort, among other factors.

Grading. Each reading assignment is self-scored by the student as a value up to 3 points. A total of 30 points is possible. The Canvas scoring facility for the reading assignments is enabled and disabled by due date. Please record your score timely. If you fail to record your score timely, you will not receive credit for the assignment. Please do not record scores for work you have not yet completely accomplished, as doing so would constitute a breach of academic integrity.

5.3. Task 03: Biweekly Programming Assignments

Except as specified in the detailed instructions related to an assignment, have another student from the class test your binary/executable program and then use the Canvas assignment submission form to submit a compressed archive containing your project. Include the following information in the completed assignment submission form:

- Programming assignment number
- Name of the program author
- Due date of the assignment
- Date when the assignment actually was completed (not necessarily the date submitted)
- Number of hours spent in programming
- Name of the peer reviewer who tested the implementation
- List of requirements not implemented, if any
- Description of defects or problems discovered by you or the peer reviewer *
- Self-evaluation based on six-point scale (see below)

Biweekly Programming Assignments will be scored by the instructor using the "Rubric for Instructor Assessment of Biweekly Programming Assignments," below.

5.3.1. Task 03A: Build the Swing Examples

Install the NetBeans IDE. Then download and build, using NetBeans, the following Java examples from the Oracle Java Swing tutorial. Be ready to demo, on short notice, your built examples in class, using a PC you bring to class or a classroom PC. Also be ready to explain and demonstrate the process you used to build the examples. The examples are found here:


An index to all the sample projects is available here. You can use a browser-based download manager or UNIX script to download all the samples at once or you can manually download them as needed, one by one.

- How to Make Applets
- How to Use Buttons, Check Boxes, and Radio Buttons
- How to Use the ButtonGroup Component (see Radio Button demo)
- How to Use Combo Boxes
- How to Make Dialogs
- How to Use Editor Panes and Text Panes
- How to Use File Choosers
- How to Use Formatted Text Fields
- How to Make Frames (Main Windows)
- How to Use Internal Frames
- How to Use Labels
5.3.2. Task 03B: Busy Box

Using NetBeans, design and implement a busy-box application, as I call it, that demonstrates the use and operation of the following principle Java Swing components. By “use” it's meant that your application should host one or more instances of the named component. By “operation” it's meant that your program should listen for, and report, the principle messages sent by components to indicate user interactions.

- Menu
- Label
- Text Field
- Text Area
- Password Field
- Button
- Check Box and ButtonGroup
- Radio Button and Button Group
- Combo Box
- List
- File Chooser
- Dialog Box

Submit a compressed archive containing your project. As always, be ready to demonstrate your application in class.

5.3.3. Task 03C: Layouts

Using NetBeans, design and implement a Java Swing application that demonstrates the use of each of the following layout strategies or managers. You can do so by nesting panels within panels, by providing a means for the user to choose the current layout manager, or other, more creative means of your choosing. A given layout manager need not control the entire main window. It’s sufficient that it control merely a portion of the main window adequate to demonstrate, by observation, that the layout manager is correctly functioning.

- Absolute Positioning
- BorderLayout
- BoxLayout
- GridBagLayout
- CardLayout
- Tabbed Pane


As always, be ready to demonstrate your application in class.

5.3.4. Task 03D: GUI Program Design

Using NetBeans, design and implement an application that demonstrates a high degree of separation between its user interface and the elements of state and actions associated with the application’s function and processing—-that is, with the application’s “back end,” as it’s called. Your application should include (at least) two main classes or components. One component establishes the user interface; call this component A. When the user manipulates the user interface, component A sends corresponding messages to the other component; call this component B. In response, component B may initiate one or more actions, including sending a message or messages to A requesting that the user interface state be updated.
This architecture is the familiar MVC design pattern implemented in the large, just as Swing and its components themselves implement MVC in the small. The MVC pattern is helpful to designing applications that are highly testable and applications that support multiple user interfaces, perhaps because of being hosted on multiple, relatively dissimilar environments or platforms.

Submit a compressed archive containing your project. Be ready to demonstrate your application in class.

Although the term includes 15-16 weeks of instruction, there are only 4 biweekly programming assignments listed here. The Mid-Term Design Project (Task 6), the Java Swing User Interface Design (Task 7), and the Java Swing Application (Task 8)—each of which entails significant Java programming—account for the apparent discrepancy.

5.4. Task 04: Mid-Term User Interface Design Project

Design and implement a Swing user interface (not a working application but only the component layout and "navigational" functions necessary to reveal hidden user interfaces) for the following conflict simulation game as described here and as further described in the referenced magazine article and other sources:

The Naval Postcard Game and Related Game Series: SALVO!

"SALVO! is a solitaire, tactical naval combat game, covering ship vs ship action. The concept of the game is simple. Each turn, each warship will either move toward the enemy (thus closing the relative range and presenting its bow), move away from the enemy (extending the relative range and presenting its stern), or offer its broadside (and keep the range the same). Thus, the relative range, and the available guns bearing on the enemy, depends on the "Action" taken by each ship each turn. There are eight Game Tables provided (Weather, Initial Range, Fire Table, Straddle Table, Special Damage, Critical Hit, Torpedo, and End of Turn Table), which cover all the eventualities of play; each table is resolved by the roll of one to three dice, with results depending on the real life characteristics of each ship. Besides the usual ratings for firepower and armor, speed can be a crucial factor as well, as the faster ship can usually dictate the "terms of the engagement". Each game described below is a complete, stand-alone design, though you may combine them if desired. (You will note that some ships appear in more than one module; ratings of such ships are always the same... they appear thus since, as mentioned, each module is a self-contained whole, and so it was not desireable [sic] for, for instance, Prince of Wales to appear in only one game.)"

"SALVO! has been well-received, and is providing gamers with a fun way to play very quick yet historically plausible naval engagements. The system provides a surprising (considering its size!) degree of realism and strategy. Add to this the fact that it is a solitaire game, where the system "plays" the opposing side, and you've got a combination that should interest many wargamers. (Please note that the SALVO! system and ship ratings are not compatible with our other tactical naval games. While very much historically based, the SALVO! series of games are (1) solitaire, (2) smaller, (3) simpler, and (4) quicker to play.)"

Considerations:

- Be sure to state explicitly your assumptions concerning your application's user(s) and user characteristics.
- Be sure not to omit essential functionality. You're welcome to share comments concerning essential functionality via the discussion area of the Canvas course site for this course.

Game play instructions:


Additional information on the game and the related game series:


<table>
<thead>
<tr>
<th>Dimension / Criterion</th>
<th>Use of UI Components</th>
<th>Visual Layout</th>
<th>Conformance to Guidelines</th>
<th>Usability</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Beginning</td>
<td>Demonstrates limited awareness of proper use of UI components.</td>
<td>Demonstrates limited awareness of principles of good visual layout.</td>
<td>Demonstrates limited awareness of user interface guidelines.</td>
<td>Design demonstrates limited awareness of usability principles.</td>
<td>Beginning content, organization, delivery, and visual aids or handouts.</td>
</tr>
<tr>
<td>2-3: Developing</td>
<td>Occasionally demonstrates proper use of UI components.</td>
<td>Occasionally demonstrates adherence to principles of good visual layout.</td>
<td>Occasionally conforms to user interface guidelines.</td>
<td>Design affords limited usability, due to inconsistent, incomplete, or improper application of usability principles.</td>
<td>Developing content, organization, delivery, and visual aids or handouts.</td>
</tr>
</tbody>
</table>
4: Accomplished

<table>
<thead>
<tr>
<th>Dimension / Criterion</th>
<th>Use of Swing Components</th>
<th>Visual Layout</th>
<th>Conformance to Guidelines</th>
<th>Usability</th>
<th>Software Defects</th>
<th>Code Quality</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Beginning</td>
<td>Demonstrates limited awareness of proper use of Swing components.</td>
<td>Demonstrates limited awareness of principles of good visual layout.</td>
<td>Demonstrates limited awareness of user interface guidelines.</td>
<td>Design demonstrates limited awareness of usability principles.</td>
<td>Demonstrates major software defects and significant missing functionality.</td>
<td>Cryptic or non-modular code that is difficult to read.</td>
<td>Beginning content, organization, delivery, and visual aids or handouts.</td>
</tr>
<tr>
<td>2-3: Developing</td>
<td>Occasionally demonstrates proper use of Swing components.</td>
<td>Occasionally demonstrates adherence to principles of good visual layout.</td>
<td>Occasionally conforms to user interface guidelines.</td>
<td>Design affords limited usability, due to inconsistent, incomplete, or improper application of usability principles.</td>
<td>Demonstrates a few major software defects or missing functionality, but not both.</td>
<td>Somewhat readable and modular code but limited use of comments and white space, if any.</td>
<td>Developing content, organization, delivery, and visual aids or handouts.</td>
</tr>
<tr>
<td>4: Accomplished</td>
<td>Generally demonstrates proper use of Swing components.</td>
<td>Generally demonstrates adherence to principles of good visual layout.</td>
<td>Generally conforms to user interface guidelines.</td>
<td>Design affords good usability.</td>
<td>Demonstrates no major software defects. Functionally complete.</td>
<td>Highly readable, modular code leaving something to be desired in terms of comments and use of white space.</td>
<td>Accomplished content, organization, delivery, and visual aids or handouts.</td>
</tr>
<tr>
<td>5: Exemplary</td>
<td>Consistently demonstrates proper use of Swing components.</td>
<td>Consistently demonstrates adherence to principles of good visual layout.</td>
<td>Consistently conforms to user interface guidelines.</td>
<td>Creative/innovative design affords high usability.</td>
<td>Demonstrates no apparent software defects. Functionally complete.</td>
<td>Highly readable, modular code with good use of comments and white space.</td>
<td>Exemplary content, organization, delivery, and visual aids or handouts.</td>
</tr>
</tbody>
</table>

Note: Final score is computed as 10 times the geometric mean of the five component scores, each ranging from 1 to 5.

5.5. Task 05: Swing Application

Propose for my approval a simple GUI application, by means of UI sketch(es) and narrative description. You may, if you like, propose to complete the implementation of the mid-term project. Given approval, design and implement the application using Java Swing. *Both the interface and the implementation are required.* Be prepared to walk the class through your design and code and to respond to questions posed by the instructor and your fellow students. You may complete the project as part of a team of two or three students. However, my expectations will rise proportionally to the size of your team. Your project must make use of the best practices for design and programming as presented in class.
Note: Final score is computed as ten times the geometric mean of the seven component scores, each ranging from 1 to 5.

5.6. Task 06. Exams

The course will have two exams: a mid-term exam and a final exam. Each exam is scored by the instructor, based on a maximum credit of 20 points.

5.6.1. Task 06A: Mid-Term Exam

The mid-term exam will be a twenty-item, multiple-choice exam given using a closed-note, closed-book, timed protocol. A maximum of seventy-five minutes will be allowed to complete the exam. For the date of the mid-term exam, see the Course Schedule, below.

Any topic covered up to the date of the mid-term exam, in class or only in the assigned reading, is a potential exam topic unless explicitly excluded. The following topics are specifically excluded from the exam:

- TBD

5.6.2. Task 06B: Final Exam

The final exam will be a twenty-item, multiple-choice exam given using a closed-note, closed-book, timed protocol. A maximum of seventy-five minutes will be allowed to complete the exam. The exam will be comprehensive; that is, any topic covered at any time during the term, in class or only in the assigned reading, is a potential exam topic unless explicitly excluded. The exclusions applicable to the mid-term exam are also applicable to the final exam.

The final exam will be given a week or two before the final-exam period scheduled by the Registrar. For the date of the final exam, see the Course Schedule, below. This is necessary in order to be able to complete most course grading while the course is still in session and face-to-face contact occurs regularly and with relative ease. Concern for student privacy arising in part due to FERPA discourages use of e-mail to communicate information such as course grades, because the perception is that e-mail is an unreliable medium as respects confidentiality. By completing the grading of most major assignments while it's still relatively easy to communicate other than by e-mail, I can hope to minimize significant grading errors and their negative effects on administrative costs, which are a significant driver of tuition.

Even though the final exam will have been given on a different date, attendance during the exam period scheduled by the Registrar is expected because that time will be used for other course-related purposes. The time allotted for the scheduled final exam session is a necessary component of this course, which requires, according to accreditation guidelines, a set minimum number of contact hours justifying its three units of assigned credit.

5.7. Task 07. Special Project

Only students seeking to earn a grade of A are required to submit a special project. Choose a project (see below), which must at least broadly pertain to the subject matter of this course, and submit it to me for approval, ideally by the end of the fifth week of the term. Once you have approval, complete, submit, and present your project. The scope of the project and the deliverables must be agreed by you and me on or before week 6 of the term. Except with my prior approval, special projects must be completed individually rather than as part of a group. The time available for your presentation will be 15 minutes. A satisfactory project will generally entail not less than 10 hours of work outside class and, depending on the topic and individual productivity, may take substantially more.

Your project may consist of any of the following:

- A presentation on an approved topic
- A presentation on a paper or book chapter taken from the textbook's bibliography or other approved source
- Other approved project

A satisfactory presentation will entail assimilating and presenting information beyond that contained in the assigned course readings. You may find the Bibliography section of your textbooks and of this syllabus helpful in locating relevant information.

Your special project will be assessed using the standard rubric for assessing presentations (see below, under Grading). Your project will also be assessed with respect to the substance of what you did or accomplished. It should be understood that, in doing a special project, you are seeking to earn an A grade. Therefore, the sophistication and depth of content presented should be commensurate with the high quality of excellence associated with a grade of A. More specifically, outstanding effort, significant achievement, and personal improvement should be clearly evident. Some measure of remarkable skill, creativity, or energy should also also evident.

Because the form taken by a special project may vary, the materials available for submission may vary. In every case, however, at least the following two components should be available:

1. Presentation materials, such as graphics and notes
2. Submission form (see below)

Submit your presentation materials and other documentary work products via Canvas, in the form of a standard zip file. If program code, whether source or executable, is part of your submission, consult the instructions for submitting programming assignments and conform to these to the degree possible. Your professor may want to compile or execute your code so the code should be submitted in a way that makes it not only possible, but as convenient as possible, to do so. For example, the professor should not be left wondering what commands are necessary to compile your program or what environment and tools are required to compile or execute it. Where possible, submit machine-readable documents but where this is not possible you may submit optically scanned documents instead.

Separately submit the following submission form as a text document or PDF document. The submission form must include the following information:

- Heading: “Special Project Submission Form”
- Course number and name
- Term
- Your name
- Title/name of your special project
- Purpose or goal of your special project (one paragraph)
- Success criterion (that is, what did you learn or accomplish and how did you determine whether your project succeeded or failed; one paragraph)
- Activities and effort (explain what steps you performed and how much time you spent on each)
- Total time spent
- Files included in your submission. What they mean/how they can be used.
- Your best argument justifying a grade of "A" (any desired length; be sure to address the criteria mentioned under Assessment, above; emphasize the value of your special project to you and to your fellow students)

**Grading of the Special Project.** The Special Project is a pass/fail assignment worth zero points. Credit for the Special Project will be made available to the student if the instructor, acting in his reasonable discretion, determines that the report offered in the student's presentation is consistent with the requirements and credibly true.

6. Learning Outcomes

6.1. Learning Themes in This Course

- Analyze a user interface context (consisting of one or more user types and one or more tasks/activities) and choose an appropriate type of user interface (Fulfilled by classroom interaction and Tasks 2, 4, 5, 6).
- Describe and explain the user interface design process (Fulfilled by classroom interaction and Tasks 2, 4, 5, 6).
- Explain the MVC (model-view-controller) design pattern and its importance to sound user interface software design and implementation (Fulfilled by classroom interaction and Tasks 2, 4, 6).
- Given a user interface context (consisting of one or more user types and one or more tasks/activities) and a user interface design, analyze and evaluate the design (Fulfilled by Tasks 2, 4, 5, 6).
- Given a user interface context (consisting of one or more user types and one or more tasks/activities) design an appropriate and effective graphical user interface based on Java Swing (Fulfilled by Tasks 2, 4, 5, 6).
- Given examples of unsatisfactory user interface design explain how the observed problems could have been avoided by following sound user interface design principles (Fulfilled by Task 2, 4, 5, 6).
- Identify and define key terms related to user interfaces and user interface design and implementation (Fulfilled by classroom interaction and Tasks 2, 3, 6).
- Identify and describe common abstract user interface components, such as radio buttons and group boxes (Fulfilled by classroom interaction and Tasks 2, 3, 6).
- Identify and describe principal Java Swing classes used to realize common user interface components (Fulfilled by classroom interaction and Tasks 2, 3, 6).
- Identify and describe various types of computer users and computer use contexts (Fulfilled by classroom interaction and Tasks 2, 6).
- Identify and describe various types of user interfaces (Fulfilled by classroom interaction and Tasks 2, 6).
- Use modern software development tools, such as the Eclipse and NetBeans IDEs, to implement user interfaces based on Java Swing (Fulfilled by Tasks 3, 4, 5).
- Design and complete a directed study project intended to develop and improve meta-learning skills pertaining to user interface design and programming (Fulfilled by Task 7). Note: This outcome is appropriate only to students seeking to earn an "A" grade in this course.

6.2. Course Alignment with Program Learning Outcomes

**CSCI 335 User Interface Design & Implementation:** This upper-division course is a core course required of all Computer Science and Computer Information Systems majors, designed to be taken after completion of freshman-level courses. Successful completion of this course (see next section) prepares students to demonstrate a Developing Proficiency toward the accomplishment of the following Program Learning Outcomes:
• **Learning Outcome 1:** Analysis, modeling, and problem solving. Students have the logical, algorithmic, and mathematical capability to model and analyze real-world problems in different application domains, to devise the problem-solving schemes accordingly, and to validate the correctness and effectiveness of the schemes.

Learning Outcome 2: Foundational knowledge and practice of computing. Students have a solid understanding of the theoretical, the operational, and the implementation underpinnings of the modern computing infrastructure to be able to effectively utilize the whole spectrum of the modern computing infrastructure, including computer hardware, software, programming environments, operating systems, and networking environments.

• **Learning Outcome 3:** Programming and system integration. Students are capable of programming using mainstream programming languages, can conduct fine software-engineering practices to implement problem-solving schemes as correct, efficient, and well-structured programs, and can integrate the programs into the computing infrastructure as functional information systems.

• **Learning Outcome 4:** Interdisciplinary competency. Students are knowledgeable in important related subject areas in business, engineering, science, or mathematics, and are capable of conducting interdisciplinary work by applying their knowledge and skills in computer science to these domains or applying their knowledge and skills in these domains to computer science.

6.3. Course Objectives and Student Learning Outcomes

6.3.1. IDEA Objective #1: Gaining factual knowledge (terminology, classifications, methods, trends).

**Important emphasis**

• Student learning outcomes (the learner will demonstrate that he or she has satisfactorily fulfilled IDEA Objective #1 by being able to):
  ○ Identify and define key terms related to user interfaces and user interface design and implementation (Fulfilled by classroom interaction and Tasks 2, 3, and 6).
  ○ Identify and describe various types of computer users and computer use contexts (Fulfilled by classroom interaction and Tasks 2 and 6).
  ○ Identify and describe various types of user interfaces (Fulfilled by classroom interaction and Tasks 2 and 6).
  ○ Describe and explain the user interface design process (Fulfilled by classroom interaction and Tasks 2, 4, 5, and 6).
  ○ Identify and describe common abstract user interface components, such as radio buttons and group boxes (Fulfilled by classroom interaction and Tasks 2, 3, and 6).
  ○ Identify and describe principal Java Swing classes used to realize common user interface components (Fulfilled by classroom interaction and Tasks 2, 3 and 6).

6.3.2. IDEA Objective #2: Learning fundamental principles, generalizations, or theories.

**Important emphasis**

• Student learning outcomes (the learner will demonstrate that he or she has satisfactorily fulfilled IDEA Objective #2 by being able to):
  ○ Explain the MVC (model-view-controller) design pattern and its importance to sound user interface software design and implementation (Fulfilled by classroom interaction and Tasks 2, 4 and 6).
  ○ Analyze a user interface context (consisting of one or more user types and one or more tasks/activities) and choose an appropriate type of user interface (Fulfilled by classroom interaction and Tasks 2, 4, 5, and 6).

6.3.3. IDEA Objective #3: Learning to apply course material (to improve thinking, problem solving, and decisions).

**Important emphasis**

• Student learning outcomes (the learner will demonstrate that he or she has satisfactorily fulfilled IDEA Objective #3 by being able to):
  ○ Given examples of unsatisfactory user interface design explain how the observed problems could have been avoided by following sound user interface design principles (Fulfilled by classroom interaction and Tasks 2, 4, 5, and 6).
  ○ Given a user interface context (consisting of one or more user types and one or more tasks/activities) and a user interface design, analyze and evaluate the design (Fulfilled by classroom interaction and Tasks 2, 4, 5 and 6).

6.3.4. IDEA Objective #4: Developing specific skills, competencies, and points of view needed by professionals in the field most closely related to this course (that is, user interface design and implementation).

**Important emphasis**

• Student learning outcomes (the learner will demonstrate that he or she has satisfactorily fulfilled IDEA Objective #4 by being able to):
  ○ Use modern software development tools, such as the Eclipse and NetBeans IDEs, to implement user interfaces based on MVC principles.
on Java Swing (Fulfilled by Tasks 3, 4 and 5).
- Given a user interface context (consisting of one or more user types and one or more tasks/activities) design an appropriate and effective graphical user interface based on Java Swing (Fulfilled by Tasks 2, 4, 5 and 6).
- Design and complete a directed study project intended to develop and improve meta-learning skills pertaining to user interface design and programming (Fulfilled by Task 9).

Note: This outcome is appropriate only to students seeking to earn an "A" grade in this course.

7. Grading

7.1. Standard Rubrics

7.1.1. Rubric for Instructor Assessment of Biweekly Programming Assignments

Note: This is essentially the same scale authored and used by Prof. Lin. Partial point values, such as 14, are permissible. Be ready to demonstrate your application in class.

Assessment:

<table>
<thead>
<tr>
<th>Points Earned</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Nothing substantial was submitted.</td>
</tr>
<tr>
<td>8</td>
<td>The submitted source code failed to compile successfully.</td>
</tr>
<tr>
<td>16</td>
<td>The submitted source code compiled cleanly but the program exhibited serious defects or was shown to be missing a couple of key features.</td>
</tr>
<tr>
<td>24</td>
<td>The submitted source code compiled cleanly and the program was shown to implement most of the required features but it also exhibited many minor defects or was shown to omit a key feature.</td>
</tr>
<tr>
<td>32</td>
<td>The submitted source code compiled cleanly and the program exhibited all the required features but also exhibited a couple of minor defects.</td>
</tr>
<tr>
<td>40</td>
<td>The submitted source code compiled cleanly and the program was shown to implement all the required features. No defects, even minor defects, were exhibited.</td>
</tr>
</tbody>
</table>

A maximum of 40 points can be scored; an additional 4 points can be scored for an exceptionally well-designed program that is exceptionally clear in its implementation. In this context, the term "exceptionally" is intended to refer to the top 5% of programs assessed since inception. It should not be expected that extra credit will necessarily be awarded to any submission in a given class.

Note: Credit may be deducted for late submission, as provided elsewhere in this syllabus and the associated policies.

7.1.2. Rubric for Instructor Assessment of Oral Communication

<table>
<thead>
<tr>
<th>Capstone 4</th>
<th>Milestone 3</th>
<th>Milestone 2</th>
<th>Benchmark 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organization</strong></td>
<td>Organizational pattern (specific introduction and conclusion, sequenced material within the body, and transitions) is clearly and consistently observable and is skillful and makes the content of the presentation cohesive.</td>
<td>Organizational pattern (specific introduction and conclusion, sequenced material within the body, and transitions) is clearly and consistently observable within the presentation.</td>
<td>Organizational pattern (specific introduction and conclusion, sequenced material within the body, and transitions) is intermittently observable within the presentation.</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Language choices are imaginative, memorable, and compelling, and enhance the effectiveness of the presentation. Language in presentation is appropriate to audience.</td>
<td>Language choices are thoughtful and generally support the effectiveness of the presentation. Language in presentation is appropriate to audience.</td>
<td>Language choices are mundane and commonplace and partially support the effectiveness of the presentation. Language in presentation is appropriate to audience.</td>
</tr>
<tr>
<td><strong>Delivery techniques</strong></td>
<td><strong>Delivery techniques</strong></td>
<td><strong>Delivery techniques</strong> (posture, gesture, eye)</td>
<td><strong>Delivery techniques</strong> (posture, gesture, eye)</td>
</tr>
</tbody>
</table>

https://compsci-wm1.biola.edu/moin/csci335?action=print
### Delivery

(posture, gesture, eye contact, and vocal expressiveness) make the presentation compelling, and speaker appears polished and confident.  
contact, and vocal expressiveness) make the presentation interesting, and speaker appears tentative.  
contact, and vocal expressiveness) detract from the understandability of the presentation, and speaker appears uncomfortable.

### Supporting Material

A variety of types of supporting materials (explanations, examples, illustrations, statistics, analogies, quotations from relevant authorities) make appropriate reference to information or analysis that significantly supports the presentation or establishes the presenter's credibility/authority on the topic.

Supporting materials (explanations, examples, illustrations, statistics, analogies, quotations from relevant authorities) make appropriate reference to information or analysis that generally supports the presentation or establishes the presenter's credibility/authority on the topic.

Insufficient supporting materials (explanations, examples, illustrations, statistics, analogies, quotations from relevant authorities) make reference to information or analysis that minimally supports the presentation or establishes the presenter's credibility/authority on the topic.

### Central Message

Central message is compelling (precisely stated, appropriately repeated, memorable, and strongly supported.)

Central message is clear and consistent with the supporting material.

Central message is basically understandable but is not often repeated and is not memorable.

Central message can be deduced, but is not explicitly stated in the presentation.

### Glossary

The definitions that follow were developed to clarify terms and concepts used in this rubric only.

- **Central message:** The main point/thesis/"bottom line"/"take-away" of a presentation. A clear central message is easy to identify; a compelling central message is also vivid and memorable.
- **Delivery techniques:** Posture, gestures, eye contact, and use of the voice. Delivery techniques enhance the effectiveness of the presentation when the speaker stands and moves with authority, looks more often at the audience than at his/her speaking materials/notes, uses the voice expressively, and uses few vocal fillers ("um," "uh," "like," "you know," etc.).
- **Language:** Vocabulary, terminology, and sentence structure. Language that supports the effectiveness of a presentation is appropriate to the topic and audience, grammatical, clear, and free from bias. Language that enhances the effectiveness of a presentation is also vivid, imaginative, and expressive.
- **Organization:** The grouping and sequencing of ideas and supporting material in a presentation. An organizational pattern that supports the effectiveness of a presentation typically includes an introduction, one or more identifiable sections in the body of the speech, and a conclusion. An organizational pattern that enhances the effectiveness of the presentation reflects a purposeful choice among possible alternatives, such as a chronological pattern, a problem-solution pattern, an analysis-of-parts pattern, etc., that makes the content of the presentation easier to follow and more likely to accomplish its purpose. 
- **Supporting material:** Explanations, examples, illustrations, statistics, analogies, quotations from relevant authorities, and other kinds of information or analysis that supports the principal ideas of the presentation. Supporting material is generally credible when it is relevant and derived from reliable and appropriate sources. Supporting material is highly credible when it is also vivid and varied across the types listed above (e.g., a mix of examples, statistics, and references to authorities). Supporting material may also serve the purpose of establishing the speakers credibility. For example, in presenting a creative work such as a dramatic reading of Shakespeare, supporting evidence may not advance the ideas of Shakespeare, but rather serve to establish the speaker as a credible Shakespearean actor.

Adapted from an "Oral Communication" rubric published by the AACU (the Association of American Colleges and Universities)  
Rev. 08/2014

The maximum possible score is 20. The score will be scaled as in the section titled "Computation of Final Grade."

### 7.2. Computation of Final Grade

#### 7.2.1. Computation of Maximum Possible Points

The summary of possible points for grade calculation are given in the table, below.

<p>| Task 02: Weekly Reading Assignments | 30 |</p>
<table>
<thead>
<tr>
<th>Task 03: Biweekly Programming Assignments</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 03A: Build the Swing Examples</td>
<td>40</td>
</tr>
<tr>
<td>Task 03B: Busy Box</td>
<td>40</td>
</tr>
<tr>
<td>Task 03C: Layouts</td>
<td>40</td>
</tr>
<tr>
<td>Task 03D: GUI Program Design</td>
<td>40</td>
</tr>
<tr>
<td>Task 04: Mid-Term User Interface Design Project</td>
<td>50</td>
</tr>
<tr>
<td>Presentation</td>
<td>20</td>
</tr>
<tr>
<td>Task 05: Swing Application</td>
<td>50</td>
</tr>
<tr>
<td>Presentation</td>
<td>20</td>
</tr>
<tr>
<td>Task 06. Exams</td>
<td>20</td>
</tr>
<tr>
<td>Task 06A: Mid-Term Exam</td>
<td>20</td>
</tr>
<tr>
<td>Task 06B: Final Exam</td>
<td>20</td>
</tr>
<tr>
<td><strong>Max. Poss. Pts.</strong></td>
<td><strong>370</strong></td>
</tr>
</tbody>
</table>

**Letter Grades**

By university policy final course grades are submitted to the Registrar as letter grades. The following table shows the number of earned points necessary to earn each possible letter grade. Note that the Registrar does not accept submission of the grade "A+.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Pts. Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>344*</td>
</tr>
<tr>
<td>A-</td>
<td>333*</td>
</tr>
<tr>
<td>B+</td>
<td>321</td>
</tr>
<tr>
<td>B</td>
<td>307</td>
</tr>
<tr>
<td>B-</td>
<td>296</td>
</tr>
<tr>
<td>C+</td>
<td>284</td>
</tr>
<tr>
<td>C</td>
<td>270</td>
</tr>
<tr>
<td>C-</td>
<td>259</td>
</tr>
<tr>
<td>D+</td>
<td>247</td>
</tr>
<tr>
<td>D</td>
<td>233</td>
</tr>
<tr>
<td>D-</td>
<td>222</td>
</tr>
</tbody>
</table>

*Note that, in order to receive a grade of A- or A a student must satisfactorily complete Task 5 Special Project. No student who has not satisfactorily completed a Special Project may earn a grade higher than B+.

A score insufficient to earn a "D-" shall be submitted as an "F" except in the context of unsatisfactory attendance, in which case a "UW" shall be submitted. A "UW" indicates an unofficial withdrawal; it has the same effect on GPA as an "F" grade.

Note: Credit may be deducted for late submission. See the subsection titled "Assignments" in the section of this syllabus titled "Policies."

**8. Course Schedule & Due Dates**

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Day of Week</th>
<th>Reading</th>
<th>Assignment</th>
<th>Exam</th>
<th>Projects</th>
<th>Special Project</th>
<th>Topic(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>08/28/2014</td>
<td>THU</td>
<td></td>
<td>Course Overview</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>09/02/2014</td>
<td>TUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>09/04/2014</td>
<td>THU</td>
<td>R01</td>
<td>Design for Interactivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>09/09/2014</td>
<td>TUE</td>
<td></td>
<td>Task 03A: Build the Swing Examples</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Day</td>
<td>Section</td>
<td>Topic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>09/16/2014</td>
<td>TUE</td>
<td>R02</td>
<td>Task 03B: Busy Box</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The Design Process I Android Architecture, Programming, &amp; Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>09/18/2014</td>
<td>THU</td>
<td>R03</td>
<td>The Design Process II Application Structure, Example Project (Yamba), UI Basics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>09/23/2014</td>
<td>TUE</td>
<td></td>
<td>Design Principles I Fragments, Intents, the Action Bar, Services, Content Providers, Lists &amp; Adapters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>09/25/2014</td>
<td>THU</td>
<td>R04</td>
<td>Task 03C: Layouts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Design Principles I Fragments, Intents, the Action Bar, Services, Content Providers, Lists &amp; Adapters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>09/30/2014</td>
<td>TUE</td>
<td></td>
<td>Design Principles II Receivers, Widgets, Networking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/02/2014</td>
<td>THU</td>
<td>R05</td>
<td>Design Principles II Receivers, Widgets, Networking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>10/09/2014</td>
<td>THU</td>
<td>MID-TERM</td>
<td>MID-TERM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/07/2014</td>
<td>TUE</td>
<td></td>
<td>MID-TERM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/14/2014</td>
<td>TUE</td>
<td></td>
<td>MID-TERM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/21/2014</td>
<td>TUE</td>
<td></td>
<td>MID-TERM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/23/2014</td>
<td>THU</td>
<td>R06</td>
<td>Design Principles III Animation Tools &amp; Basics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10/28/2014</td>
<td>TUE</td>
<td></td>
<td>Task 04: Mid-Term User Interface Design Project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10/30/2014</td>
<td>THU</td>
<td>R07</td>
<td>Task 04: Mid-Term User Interface Design Project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Design Principles IV The Android Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11/04/2014</td>
<td>TUE</td>
<td></td>
<td>Design Principles IV The Android Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11/06/2014</td>
<td>THU</td>
<td>R08</td>
<td>Design Principles V Skeleton Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11/11/2014</td>
<td>TUE</td>
<td></td>
<td>Design Principles V Skeleton Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11/13/2014</td>
<td>THU</td>
<td>R09</td>
<td>Advanced</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Day</td>
<td>Topic</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-----</td>
<td>-------</td>
<td>-------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11/18/2014</td>
<td>TUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11/20/2014</td>
<td>THU</td>
<td>R10</td>
<td>Advanced Topics II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11/25/2014</td>
<td>TUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>THANKSGIVING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12/02/2014</td>
<td>TUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12/04/2014</td>
<td>THU</td>
<td></td>
<td>FINAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>12/09/2014</td>
<td>TUE</td>
<td></td>
<td>Task 05: Swing Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12/11/2014</td>
<td>THU</td>
<td></td>
<td>Task 05: Swing Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Exam Period</td>
<td>12/16/2014 3:00-5:00 p.m</td>
<td>TUE</td>
<td></td>
<td>Task 07: SPECIAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>