
















Course	SP19 Est.	Wk.	Sess.	Session description	Resources	Learning objectives	Out-of-class work
CIT-130	WED 17-APR-19						
CIT-130	MON 22-APR-19	12	1		<b>GUI &amp; Inheritance project workshop</b>  AWT GUI Intro packet  LIANG9 Textbook: Chapter 12: GUI basics	12.L.1: 12.L.2:	
CIT-130	WED 24-APR-19		2	 Course planning survey  LIANG9 Textbook: Chapter 16: Event-driven programming  LIANG9 Textbook: Chapter 17: GUI Components   Movie GUIs			
CIT-130	MON 29-APR-19	13	1		<b>Exceptional term projects: design and workshop time</b> 	13.L.1: 13.L.2:	
CIT-130	WED 1-MAY-19		2				
CIT-130	WED 8-MAY-19	14	1	* Bring fully-baked projects to share. * Same time and place as normal Wednesday class	<b>Sharing term projects and final checkout</b>  Course planning survey  Final session checklist	14.L.1:	




## CIT-244: Object-oriented design in Java








The following table maps course session dates, lesson topics, LIANG9 references, and content links for all three Java courses in the series.

18-FEB-19







Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
CIT-244		1	1		<b>String manipulation and array review</b>  1. Introduction to object-oriented design	<b>CCAC.244.LO.1:</b> Apply Java language elements to use string(sic) processing techniques in a program  <b>CCAC.244.LO.2:</b> Apply Java language elements to create programs with single dimension arrays of primitives and	





Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
CIT-244	WED 4-SEP-19		2		2. Exploring past project code from last term 3. Digging into password strength checking   <a href="#">Week 1 Module</a>	objects. Apply Java language elements to use <code>String</code> processing techniques in a program   <b>LIANG9 Textbook:</b> Chapter 6: Arrays Chapter 9: Strings	
CIT-244	MON 9-SEP-19	2	1		<b>Inheritance revisited: Teasing apart use cases for plain inheritance, abstract methods, and interfaces</b>	2.L.1: 2.L.2:   <b>LIANG9 Textbook:</b> Chapter 15: 15.1 - 15.8	
CIT-244	WED 11-SEP-19		2		2.L.1: 2.L.2:  		
CIT-244	MON 16-SEP-19	3	1		<b>Exploring constructors and interfaces with a Comparable Computer object</b>   <a href="#">Week 3: Abstract classes &amp; interfaces</a>  <b>Exercise 1:</b> Group code-along of an <code>Integer</code> array which has a natural ordering and can be sorted by <code>Arrays.sort(Object[] o)</code>  <b>Exercise 2:</b> Creating the <code>Computer</code> class and a four-argument constructor. Populating an array with <code>Computers</code> .  <b>Exercise 3:</b> Implement <code>Comparable</code> inside <code>Computer</code> and test with <code>Arrays.sort()</code> .  <b>Exercise 4:</b> Abstract classes - Add a child	3.L.1: 3.L.2:   <b>LIANG9 Textbook:</b> Chapter 11: Inheritance and polymorphism Chapter 15: Abstract classes and interfaces	
CIT-244	WED 18-SEP-19		2				
CIT-244	MON 23-SEP-19	4	1		<b>Inheritance Practice: Modeling computer timelines and file I/O</b>  Study this document used by the CIT-115 students to create the computer history	Write to and from files  Create a data encoding scheme for the data in files that works with your program's inheritance scheme	

Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
					<p>timeline in the hall. Your job is to model the timeline components in Java such that the data is stored in a File and can be retrieved and re-created as the timeline evolves.</p> <p> <a href="#">Computing power timeline</a></p> <p> <a href="#">LIANG9: Chapter 11 - Inheritance</a>  <a href="#">LIANG9: Chapter 15 - Abstract classes</a>  <a href="#">LIANG9: Chapter 14.10-14.13 - File i/o</a></p> <p> <a href="#">File I/O Project specs</a></p>		
CIT-244	WED 25-SEP-19		2				
CIT-244	MON 30-SEP-19	5	1	Creating class diagrams for Timeline project	<p><b>Data store API project design and workshop time</b></p> <p><b>Connecting Java applications and the file system: Introducing file I/O</b></p> <ol style="list-style-type: none"> <li>1. Install the EasyUML diagram plugin in NetBeans</li> <li>2. Create a new project of type UML</li> <li>3. Create an empty class in some other project and then click and drag it into your EasyUML project</li> <li>4. With one class inside the UML project, right click that class and select "easyUML generate class diagram" to create a new diagram</li> <li>5. Tweak and refine your UML diagram using the options in the palette for tools</li> <li>6. You can generate the Java from your diagram by selecting any class diagram over in your projects tree and selecting "easyUML generate code". You'll specify the project in which you'd like easyUML to dump the</li> </ol>	<p>Write to and from files</p> <p>Create a data encoding scheme for the data in files that works with your program's inheritance scheme</p>	

Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
CIT-244	WED 2-OCT-19		2	Implement file I/O with text-based UI for timeline project	<p>classes it generates. (It will place them all in the default package for that project. You'll want to them move those class files into their appropriate named package)</p> <p> <a href="#">File I/O Project specs</a></p> <p> <a href="#">LIANG9: Chapter 11 - Inheritance</a>  <a href="#">LIANG9: Chapter 15 - Abstract classes</a>  <a href="#">LIANG9: Chapter 14.10-14.13 - File i/c</a></p>		
CIT-244	MON 7-OCT-19	6	1	<p><b>DUE AT BEG OF CLASS: Text-based UI for timeline project with reading and writing from files for persistence</b></p> <p>Refactoring to align to the MVC framework</p>	<p><b>Implementing the MVC framework</b></p>	<p>6.L.1: 6.L.2:</p> <p> LIANG9 Textbook:</p>	
CIT-244	WED 9-OCT-19		2	Building new functionality with minimal changes: All Hail the MVC design framework			
CIT-244	MON 14-OCT-19	7	1	Migration to GUIs from our text-based interface	<p><b>Migration to GUIs</b></p> <p> <a href="#">Oracle's Master Swing Tutorial</a>   <a href="#">Oracle's Swing example index</a>   <a href="#">LIANG9 Textbook: chapter 12 GUI Basics</a></p>	<p>7.L.1: 7.L.2:</p>	
CIT-244	WED 16-OCT-19		2	Continued GUI migration and event handling			
CIT-244	MON 21-OCT-19	8	1	Inner class worksheet and review of Listener methods	<p><b>Peer-testing the GUI and creating feature requests</b></p> <p>Prepare for CIT-115 students testing your program on Wednesday, 20 March</p> <p></p>	<p>8.L.1: 8.L.2:</p>	
CIT-244	WED 23-OCT-19		2	Continued GUI building for Timeline project			

Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out of class work
CIT-244	MON 28-OCT-19	1		<p>Revised: DUE AT BEG OF CLASS: Working GUI that takes in timeline data, writes to a file, and reads file data back in, creating an organized timeline</p> <p>During class we will prep the sharing of the programs with CIT-115, so please be ready to package and share.</p>	<p><b>Preping for sharing</b></p> <p>Please complete the following steps for Monday, 1 APR at 8:00 am (when CIT-115 meets)</p> <ol style="list-style-type: none"> <li>1. Complete your GUI to a sharable state (no obvious bugs or silly stuff dangling all over the place)</li> <li>2. Push your code to your github repo</li> <li>3. Make an entry in our <a href="#">upload log so we can find your code</a></li> </ol>	<p>9.L.1:</p> <p>9.L.2:</p>	
CIT-244	WED 30-OCT-19	9	2	<p>Review client feedback on GUI applications in CIT-115</p>	<ol style="list-style-type: none"> <li>4. Create a markdown file called readme.md in the same directory as your code and include the following content. <a href="#">You can use this great markdown guide</a> to help you write the markdown. <ul style="list-style-type: none"> <li>◦ Headers</li> <li>◦ 1 Image</li> <li>◦ 1 Link</li> <li>◦ 1 Code example with explanation</li> <li>◦ 1 review question for the 115 student to write a response to</li> <li>◦ Step-by-step on how to run your program from your git hub (include how to clone your repo, and how to open your project, and which class to run)</li> </ul> </li> </ol>		

Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
CIT-244	MON 4-NOV-19	10	1	Review feedback from CIT-115 students Group coding of Merge sort	<b>Algorithm development: Sorting!</b>  <a href="#">Sorting project</a> Small-group exploration of sorting algorithms 1. Bubble sort 2. Merge sort 3. Quick sort	10.L.1: 10.L.2:	
CIT-244	WED 6-NOV-19		2	Introduce sorting comparison project and divide into groups	4. Heap sort (binary tree) 5. Bucket sort   <a href="#">Sorting visualization</a>  <a href="#">LIANG9 Textbook: Chapter 25</a>		
CIT-244	MON 11-NOV-19	11	1	Sorting metrics continued	<b>Sorting</b>  <a href="#">Sorting project</a>  <a href="#">Sorting visualization</a>  <a href="#">LIANG9 Textbook: Chapter 25</a>  <b>Project wrap-up TODO</b> 1. Write javadoc comments for each method in both of your classes using the <a href="#">javadoc doc as your guide</a> 2. Push your code up to your github repository. Create a small readme.md for documenting what the code does 3. Make an entry in our <a href="#">upload log so we can find your code</a>	11.L.1: 11.L.2:	

Course	SP19 Est.	Wk.	Sess.	Session description	Module links	Language objectives	Out-of-class work
CIT-244	WED 13-NOV-19		2		4. Clone a peer's repo and test their sorting algorithms using your test class and its various test methods. Write a note card evaluating the correctness and efficiency and beauty of your peers' code. Show your peer and your instructor your note card.		
CIT-244	MON 18-NOV-19		1		<p><b>Sort algorithm testing &amp; git</b></p> <ol style="list-style-type: none"> <li>Finalize sorting algorithm class (see project link below)</li> <li>Create a git branch for just your sorting algorithms</li> <li>Push your finalized work to remote repo</li> <li>Clone peer's repo to your system</li> <li>Compare test results using your own and others client methods</li> <li>Refactor peer's code and push to your own remote</li> <li>Create a <a href="#">pull request on the original author's repo</a> and merge changes in a new branch</li> <li>Merge in objects from a pull request to one's own rep</li> </ol>		
CIT-244	WED 20-NOV-19	12	2	<p> <a href="#">Course planning survey</a></p>	<p><a href="#">AZ Sorting project</a></p> <p> <a href="#">Git Cheat Sheet by Atlassian</a></p> <p> <a href="#">LIANG9 Textbook: Chapter 25, Sorting</a></p> <p> <a href="#">CIT-244 Final project specs</a></p>		
CIT-244	MON 25-NOV-19		-				