[CSC110 Syllabus] | [CIS163 Syllabus] | [ General Information ] | [Schedule] | [Instructor]

# Spring 2005 Schedule
4:00 PM Wednesday

The following is a schedule of class topics, handouts, assignments and exam dates. I will be adding and changing this schedule throughout the semester to meet class needs. Items that are blank are still under review. ***This schedule is subject to change and most likely will***.

**The first line of all your programs should contain your name, the exercise and the day/time your class meets. Example: // Chris Goodstudent, Ex 2.5, W 4:00**

| | Topics Covered | Assignments and Handouts |
|---|---|---|
| **1**<br>Jan 19 | • Introductions<br>• Review syllabus<br>• Setting up User ID and PW<br>• Blackboard email setup<br>• In-class group assignment: *Navigating the Class Web Site*<br>• Hand out **java install disk**<br>• Textpad and a first java program<br>• Tour lab - B123 | **In Class Activities/Handouts:**<br>• Blackboard account setup<br>• Set up your Blackboard email - See announcement on course Blackboard site. For further help read : Setting up your Blackboard email<br>• Navigating the Class Web Site<br>• Using Textpad to write, compile and run your Java applets<br>• Campus Map and Building B Map. The computer lab is located in the B building (B123)<br>**Homework (due next class meeting):**<br>• Ex 2.3, 2.4 (did you add your name, etc to the first line?)<br>• Print out a check sheet from your program of study and highlight this course. If you are not in a specific program, then give a brief statement of why you are taking this course.<br>• Are you smiling?<br>**Reading:**<br>• Ch 1, 2, 3 |

| | | |
|---|---|---|
| **2**<br>Jan 26 | • A little Java History<br>• Programming concepts<br>• Intro to graphics<br>• Statement execution sequence<br>• Java Style Manual<br>• Commenting your code | **In Class Activities/Handouts:**<br><br>• Java History PPT<br>• From Source Code to Running Program<br>• snowman example (will post on Blackboard)<br>• Sun Java Style Manual<br>• Java Style Exercise<br>• A real mess!<br>• Homework Grading Guidelines<br><br>**Homework (due next class meeting):**<br><br>• Please review the Homework Grading Guidelines.<br>• Ex 3.5, plus enclose the entire target in a rectangle or come up with your own creative addition involving *drawstring* and color fills.<br>• Take your completed Ex 3.5 and revise it so that if follows the style manual rules<br><br>**Reading:**<br><br>• Ch 4,5 |
| **3**<br>Feb 2 | • Literals<br>• Primitive data types<br>• Assignment<br>• Arithmetic operators and expressions<br>• Operator precedence<br>• Variables<br>• Identifiers<br>• Introductory type conversion<br>• New style issues<br>• ==========<br>• Showing line numbers with textpad<br>• Developing methods<br>• Passing parameters<br>• Local variables<br>• By value<br>• Reusing methods<br>• Returning Values | **In Class Activities/Handouts:**<br><br>• Quiz #1- 10 points<br>• Java primitive data type reference<br>• int or float?<br>• Java operator precedence reference<br>• FunnyMoney Example<br>• GoalPost and using methods<br>• Corrections to errors in text (note: minus sign in some line numbers means count up from the bottom of the page)<br><br>**Homework (due next class meeting):**<br>Programs must meet standards of Style Manual Week 3<br><br>• Please review the Homework Grading Guidelines.<br>• Ex 4.3 (answer must be correct to at least one decimal place)<br>• Ex 4.7 (hint)<br>• Work along with the examples in the chapter 5 (good idea for all chapters)<br>• Ex 5.2 (hint)<br><br>**Reading:**<br><br>• Ch 6, 7 |

| | | |
|---|---|---|
| **4**<br>Feb 9 | • Local variables<br>• Returning values<br>• By val vs By Ref<br>• Understanding scope<br>• The event concept<br>• The AWT<br>• Adding scrollbars and labels to the screen using the AWT<br>• Listening for events<br>• =========<br>• Intro to selection structure<br>• if...else<br>• comparison operators<br>• (ch 7 through p 97) | **In Class Activities/Handouts:**<br><br>• Variables/Methods/Scope Review Shapes.java<br>• Quiz #2- 10 points<br>• Events in Java: FirstEvent<br>• Nested if..else Statements<br>• Boolean Logic Tables<br>• Need some additional help - check with the Tutoring Center. They can schedule an appointment with a Java tutor for you.<br><br>**Homework (due next class meeting):**<br>Programs must meet standards of Style Manual Week 3<br><br>• Please review the Homework Grading Guidelines.<br>• **CIS Students Only:** Ex 5.9 - A method that has a return value<br>• **CSC Students Only:** Ex 5.8 - A method that has a return value<br>• Ex 6.4 - Oval<br><br>Programs must meet standards in Style Manual Week 4<br><br>• Please do all the self-tests<br>• Ex 7.4 Please use a **scrollbar** not a textfield to input the age<br><br>**Reading:**<br><br>• Ch 8 pgs 116-121 |
| **5**<br>Feb 16 | • boolean operators &&, \|\|, !<br>• nesting if statements<br>• boolean variables<br>• buttons<br>• text fields<br>• ========<br>• random numbers<br>• switch<br>• intro to loops<br>• while loops<br>• =========<br>• brief description of exam 1 | **In Class Activities/Handouts:**<br><br>• Nested if..else Activity using CarFun code<br>• Event handling Activity and ButtonCount code<br>• Quiz #3: Through Ch 7 (10 points)<br>• Event handling skeleton<br>• Style Manual Week 5<br>• Exam #1 Toolkit<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 4.<br><br>• Please do all the self tests<br>• Ex 7.9 -Elevator<br>• Ex 7.2 - Cards (Use switch to convert the numbers.)<br>• Ex 7.8 Please note: this calculator doesn't |

| | | |
|---|---|---|
| | | work the way a normal calculator works. Read the description of its function carefully.(hint) . Run a demo of the calculator. Due on **class #7**.<br><br>This program must meet standards in Style Manual Week 5.<br><br>• Ex 8.2 (Use while)<br>• Study for exam.<br><br>**Reading:**<br><br>• Ch 8 pgs 122 - 133 |
| **6**<br>Feb 23 | • for loops<br>• do while loops<br>• nested loops<br>• Activity Diagrams/Flowcharting<br>• ----------<br>• Exam 1 | **In Class Activities/Handouts:**<br><br>• Self-test 8.1 - compare **while**, **for**, and **do** loops<br>• Self-test 8.6, 8.7 - nested For Loops<br>• Activity Diagrams and Flowcharting tutorial:<br>• Demo Calculator (due next class)<br>• Style Manual Week 5<br><br>**Homework (due next class meeting):**<br><br>• Continue working on Calculator<br>• 8.3 (Use for) Milky Way<br>• 8.8 (use nested for loops) Multiplication Table<br><br>**Reading:**<br><br>• Ch 9 - This chapter is fundamental to Object Oriented Programming. Please read carefully! |
| **7**<br>Mar 2 | • Nested loops revisited<br>• Review Exam 1<br>• Check Grades<br>• ----------<br>• intro to object-oriented design<br>• creating and controlling an object<br>• constructors<br>• displaying an object<br>• attributes<br>• operations on objects<br>• overloading<br>• default constructors<br>• implementing methods | **In Class Activities/Handouts:**<br><br>• Review of loop statements<br>• Exam 1 student solutions<br>• Monster Cookbook #1: Building a Class<br>• Monster Display Code<br>• Monster Compile Fix<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5. Will be graded according to Homework Grading Guidelines.<br><br>• Do all self-test questions<br>• Build a base monster class and a monster controller class as explained and demonstrated in class. The monster and the monster controller should be placed in |

| | | |
|---|---|---|
| | | separate files.<br>• Train your monster to do 5 things and enhance your controller to direct the monster to do those things. Be ready to demo your monster to the class next time<br><br>**Reading:**<br><br>• Ch 9 (again-this is a very important chapter), Ch 10 (optional), Ch 11 |
| **8**<br>Mar 9 | • Monster training session<br>• accessor methods<br>• private methods<br>• static methods, attributes<br>• garbage collection | **In Class Activities/Handouts:**<br><br>• First part of class will be monster training session<br>• Monster demos - students<br>• Begin OO concepts review. monsterOO class and monsterOO controller<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5. Will be graded according to Homework Grading Guidelines.<br><br>• Continue to train your monster to do at least 5 things and enhance your controller to direct the monster to do those things.<br>• **Complete Quiz #4 on Objects and Classes on BlackBoard located under Quizzes.**<br><br>**Reading:**<br><br>• Ch 11 (again), Inheritance Handout,<br>• **Sun Java Online Tutorial on Inheritance** |
| **Mar 16** | **Spring Break- College Closed** | |

| | | |
|---|---|---|
| **9**<br>Mar 23 | • Review Take Home Quiz<br>• OO Concept Review<br>• ======<br>• inheritance<br>• overriding<br>• final static<br>• ======<br>• **Brief Exam #2 Review** | **In Class Activities/Handouts:**<br>• More monster demos - student<br>• Review Quiz #4 and OO concepts<br>• Base monster class and monster controller for inheritance assignment<br>• Exam #2 Toolkit<br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br>• Monster inheritance assignment Hint: There is an example under Answer Keys<br>• Study for exam #2 (chs 8, 9, 11)<br>**Reading:**<br>• Review Ch 8, 9, 11 for exam |
| **10**<br>Mar 30 | • Revisit Inheritance<br>• Testing - black box<br>• Testing - white box<br>• Debugging<br>• =========<br>• Exam 2 | **In Class Activities/Handouts:**<br>• Some Student Inheritance Monster demos<br>• Inheritance Concepts Revisited<br>• Black box/white box testing problem BiggestController.java<br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br>• Finish up all Monster Assignments.<br>• 23.2 - Black and White Box Test Plans (hint)<br>**Reading:**<br>• Ch 13 (pgs 223- 242), Ch 14<br>• Ch 23, Ch 24 |
| **11**<br>Apr 6 | • Check Grades<br>• Review Exam 2<br>• ==========<br>• introduction to arrays<br>• applications of arrays<br>• Sort algorithm | **In Class Activities/Handouts:**<br>• we'll do part of the statistics array class assignment in class<br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br>• Statistics Array Class Assignment<br>**Reading:**<br>• Ch 14, Ch 12, Ch 15 |

| | | |
|---|---|---|
| **12**<br>Apr 13 | • multi-dimensional arrays<br>• =========<br>• Calculations<br>• Strings | **In Class Activities/Handouts:**<br>• Strings Quiz<br>• StringDemo<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br>• After completing the StatsArray class, create a new class called MultiArray. I have given you a starter file for MultiArray and a complete MultiArrayController. MultiArray is the same as StatsArray, except that the array it houses is two dimensional. Revise all methods to work with the two dimensional array except the sort method. You do not need to include the sort method in MultiArray. Instead, create new methods that add up the 2nd column of numbers in the multidimensional array and a method to display the 2nd column. You will see in the starter file that I have completed some of the methods to give you a hint.**You may examine how the program should work by viewing a demo of the Multidimensional Array.**<br>• Draw a class diagram for your MultiArray class.<br>• 15.1. Also, if the two strings are EQUAL, convert the joined string to all uppercase before displaying it.<br>• 15.3. Also, convert the number to an int or float as applicable. Do NOT display the converted number to the Applet window. Instead, display it to System.out.println, as described in Ch 24 on debugging.<br><br>**Reading:**<br>• Ch 16, 18 |
| **13**<br>Apr 20 | • Exceptions<br>• Java Application Examples<br>• Brief Review for Exam 3 | **In Class Activities/Handouts:**<br>• Exercise 16.1. Throw an ArithmeticException if the denominator is zero.<br>• Example of Applet to Application conversion - Start with these files: StatsArray and StatsArrayController. After converting to application : StatsArrayControllerApp.<br>• Exam #3 Toolkit |

| | | |
|---|---|---|
| | | **Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br><br>• 16.3. Here is a starting file that compiles and runs. You will need to add Exception Handling. Please check for the following exceptions:<br>  ○ **NumberFormatException** (ie. a nonnumeric character was entered)<br>  ○ **IllegalArgumentException** (thrown by checkforBadNumber - see below)<br>  ○ **Exception**, the superClass. (for some unexpected Exception)<br>  ○ Please include a method called **checkForBadNumber** that throws an IllegalArgumentException if any input numbers are zero or negative.<br>  ○ Please include **finally** statement with a statement of your choice.<br>• Convert any prior assignment to an application. (except StatsArray)<br><br>**Reading:**<br><br>• Ch 15 (pages 287-288 on StringTokenizer class)<br>• Ch 19 pages 344-53, 360-66 (files)<br>• Ch 21- OO Design |
| **14**<br>Apr 27 | • File Streams<br>• OO Design<br>• Ethics<br>• =====<br>• Exam 3 | **In Class Activities/Handouts:**<br><br>• Using the FileReader.txt file, write a java program called FileR that reads in the data and displays it using System.out.println<br>• Modify FileR to create a new program called FileRW that reads in FileReader.txt and writes the data to FileWriter.txt.<br>• Modify FileRW to use the StringTokenizer (page 287) to break the input line into fields.<br>• Using Files Checklist<br><br>• Design Example<br>• Software Engineering Code of Ethics and Professional Practice from the IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices<br>• Code of Ethics in Computing from the |

| | | |
|---|---|---|
| | | Association of Computing Machinery (optional reading)<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br><br>• Please write a class called NameSwitch which reads in a text file containing first name and last name separated by commas. Then, use System.out.println() to display each name in lastname, then firstname format. Example input "George,Bush" Example output: "Bush,George" I've created a file called names.txt which contains 10 names. You can download this file and use it with your program.<br>• Modify NameSwitch so that the reversed names are sent to an output file instead of to System.out.println()<br>• Ethics assignment. Pick a code of ethics from Software Engineering Code of Ethics and Professional Practice . Select one you believe will apply to you and read it. Now read the Case on "Who Can Change Proprietary Source Code and answer the questions at the end of each section. What guidance does the code of ethics provide that might help you resolve case? Your work must be typed and free of spelling and grammatical errors. Approximately 500 words in length.<br><br>**Reading:**<br><br>• Ch 13 pages 243-249(ArrayLists) |
| **15**<br>May 4 | • Review Exam 3<br>• ArrayLists<br>• Stacks/queues<br>•<br>• Review for Final | **In Class Activities/Handouts:**<br><br>• **Complete Course/Instructor Online Survey**<br>• ArrayList example: Will need the class definition for Artist and the controller that creates and manages the ArrayList. The controller is called HitList<br>• Final Exam Toolkit<br><br>**Homework (due next class meeting):**<br>Programs must meet standards in Style Manual Week 5.<br><br>• ArrayList exercise 13.17 (queue) or 13.18 (stack), whichever you prefer. Please use the artist objects from the Artist class definition as the objects to be queued/stacked. I also recommend |

| | | |
|---|---|---|
| | | modifying the HitList user interface code to create your stack or queue controller.<br>• Study for exam<br>**Reading:**<br>• No new reading |
| **16** Wed May 11 | • Final Exam - Comprehensive | 4:00p - 6:40p in C104. The exam is open book and notes. Please bring a pencil. |