CISS 238 Java Programming

Session (09/53)

January 11 – March 6, 2010

On-line Course Syllabus

Course Description

CISS 238: Java Programming

An introduction to programming using Java. Topics include methods, classes, objects, advanced object concepts, input, selection, repetition, arrays and strings, applets, HTML, graphics, inheritance concepts, abstract windows tool kit, file input and output.

Prerequisites:

- CISS 170 - Introduction to Computer Information Systems
- MATH 150 - College Algebra.

I. Overview and Course Goals

Welcome to Java Programming (CISS 238), online! Material in this course is highly dynamic and will require you to analyze challenges applying programming constructs to solve. The syntax of the Java language is fairly simple; however, the application of the syntax can, and will, provide a challenge to you. You will be given ample opportunity to learn each objective so that you will be prepared for the midterm and final exams as well as real world challenges.

Each week we will focus different aspects of the Java language. You will learn about objects early as Java is an object-oriented language and is the basis for most development projects.

To assist you in developing your applications, we will be using an Integrated Development Environment (IDE) to complete our assignments. This will assist you in learning the syntax of the language as well as provide you with experience using an IDE that is used professionally within the industry. However, please be aware that during the exams, you will not have access to this tool.

Computer Requirements

To participate in this course and complete the Dropbox assignments you will find it necessary to have the following software installed on your computer:

- Java Development Kit Version 1.5 or better.
- Eclipse Integrated Development Environment Version 3.1 or better

This software is located on the CD included with the book.

Weekly Information

During Week 1, we will begin our understanding of the Java language, starting with the structure of application development and why we program to classes and objects. This week will focus on the following (MLO 2, 3):

- Object Oriented Programming
- Parts of a Java Program
- Outputting to the console window
- Data types
- Arithmetic and assignment operators
- Use and understanding of the String class
- Objects
- Classes
- Constructors
- Imports
- Packages

During Week 2, we will learn how to use decision structures and loops. You will also be introduced to basic file input/output. The focus this week is as follows (MLO 1):

- if statement
- if-else statement
- if-else-if statement
- Nested if statements
- switch statement
- while loop
- do-while loop
- for loop
- Nested loops

During Week 3, we will take a closer look into objects and classes. We will also cover the use of Arrays and ArrayLists and how they can benefit you in OO programming. (MLO 2, 3)

- Static classes
- Overloading methods

- Overloading constructors
- Passing objects as arguments
- toString
- equals
- Aggregation
- this reference variable
- Inner Classes
- Arrays
- String Arrays
- Arrays of objects
- ArrayList
- Two-dimensional arrays

During Week 4, we will begin text processing using string objects and tokenizers. We will also explore wrapper classes. (MLO 2, 3)

- Wrapper classes
- Character testing and conversion
- String objects
- StringBuilder
- Tokenizing Strings

During Week 5, we will begin our study of Inheritance, polymorphism and abstract classes and methods. (MLO 3)

- Inheritance
- Superclass
- Protected members
- Object class
- Polymorphism
- Abstract classes and methods

During Week 6, we will expand our capabilities by understanding how to handle exceptions in our code and provide feedback to the user. We will also cover file input and output for reading and writing files. (MLO 4, 6)

- Handling exceptions
- Throwing exceptions
- Binary files
- Random access files
- Object serialization

During Week 7, we will start building GUI (Graphical User Interface) applications using AWT (Abstract Windowing Toolkit). (MLO 5)

- Dialog boxes
- Windows
- Radio buttons and check boxes
- Borders
- Splash screens
- Lists
- Combo boxes
- Images, labels and buttons
- File choosers
- Color choosers
- Menus
- Sliders

During Week 8, we will cover how to connect to databases using a JDBC connection and issues queries against the database. We will also have our Final Exam. (MLO 7)

- Connection
- JDBC
- executeQuery
- executeUpdate
- Statements
- ResultSet
- PreparedStatement
- Final Exam

Online-Help

If you are having difficulties with any of the projects you are welcome to contact me via email at msjenkins@cougars.ccis.edu. Please ensure that you send me a copy of the project/problem you are working on as well as the problem description from the book. This will assist me in determining exactly where you are and help you resolve your problem to put you back on track and moving forward.

Course Demo

If this is your first course online or the first time using the Desire2Learn module you will find the course demo helpful at http://www.ccis.edu/online/demo.asp.

II. Course Objectives

- To decompose and map problems onto Java syntactical constructs.
- To test Java programs using appropriate techniques.
- To utilize graphic processing techniques.
- To build graphical user interfaces.

- To utilize object-oriented tools and techniques in program construction.

CISS 238 Measurable Learning Outcomes

Upon Complete of this course students will be able to:

1. Utilize basic control structures such as if/then/else, do while, do until, while, and for/next.
2. Utilize modules, classes, and procedures to organize program structure.
   Process arrays.
3. Employ object-based constructs and techniques such as classes, inheritance, and polymorphism.
4. Explain exception handling and utilize it in programs.
5. Develop graphical user interfaces.
6. Process files and streams.
7. Connect to databases and issue queries.

III. Course Policies

View the Student Orientation web sites below for details about taking an on-line course.

- Course Demo
- How Online Learning Works
- Is Online Learning for Me?
- Technical Requirements
- Self Assessment Quiz

There will be no discrimination on the basis of sex, race, color, national origin, sexual orientation, religion, ideology, political affiliation, veteran status, age, physical handicap, or marital status. Students with documented disabilities who may need academic services for this course are **required** to register with the Coordinator for Disability Services. **Until the student has been cleared through the disability services office, accommodations do not have to be granted.** It is vital if you are a student who has a documented disability to read the entire syllabus before signing up for the course. The structure or the content of the course can make an accommodation not feasible. The policies and related syllabus matters remain subject to change in the event of extenuating circumstances.

Students with documented disabilities who may need academic services for this course are required to register with the ADA coordinator of Columbia College at 573 875 7626. Read the entire syllabus before continuing the course.

Student Conduct

The instructor reserves the right to manage a positive learning environment and thus will not tolerate inappropriate conduct in the course. All Columbia College students, whether enrolled in a land-based or online course, are responsible for behaving in a manner consistent with

Columbia College's Code of Student Conduct and Ethics Code for Computer Users. Students violating these codes will be referred to the Campus Life Office for possible disciplinary action. The Code for Student Conduct and the Ethics Code for Computer Users can be found in the Columbia College Student Handbook, a copy of which can be obtained by calling the Campus Life office.

On-Line Participation

This course is offered on-line, over the Internet, using the Internet and the World Wide Web, using publishing technology provided by Desire2Learn and Columbia College Participation on-line is expected and continuous throughout the course. Failure to turn in assignments by the date due, or timely participation in online discussions may result in the student failing the course. Emergencies should be communicated and documented to the instructor as soon as possible. Students are expected to read the assigned texts each week and login to the class conferencing, and post at least one message per week to each of the Discussion topics provided in the Discussion area. Active participation in the course will guide students in studying for the exams and in researching for the scholarship. The instructor will facilitate online discussions in the conference room by responding to posted messages. See "Ground Rules for On-line Participation" for additional information.

A class week is defined as the period of time between Monday and Sunday. The first week begins the first day of the session (**Monday, January 11, 2010**) and ends midnight the following Sunday except for Week 8 when the week and the course will end Saturday at midnight. Both conference and Dropbox assignments, scheduled for completion during a class week, should be submitted or posted by the weekly due dates stated on the grading schedule below. Dropbox assignments (projects) should be completed and successfully uploaded in the Dropbox section of your course so that they are received by the due date. NOTE: Because this is an online course designed to get feedback on assignments to you directly via Internet, you must make prior arrangements before submitting a paper via fax or the postal service. If you ever have problems transmitting your assignments to me, telephone me immediately at (303) 378-8513 and we'll get the problem solved.

Ground Rules for On-line Participation

Students should use CougarMail email for private messages to the instructor and other students and for assignments, which cannot be otherwise uploaded into the appropriate dropbox. The Class Conference is for public messages so we can see what each other has to say about any given topic, and respond to if desired.

Students are expected to participate in on-line discussions, as well as with other appropriate online activities including sending/receiving email and navigating and conducting research over the World Wide Web.

All students will observe Conventions of "online etiquette," when communicating on-line which includes courtesy to all users.

Students may get assistance with computer related problems through the instructor. For assistance with eServices or CougarMail accounts, please contact the Columbia College Helpdesk at (800) 231-2391 x4357 or CCHelpDesk@ccis.edu. For assistance with the D2L course environment, please contact the D2L Helpdesk at (877) 325-7778 or http://www.desire2learn.com/sup/user/support.asp.

Dropbox assignments will be prepared using eclipse and uploaded using the "Dropbox" section of your course. That method preserves their formatting.

All email communication from the instructor to the student will be through the College Cougars mail system. If there are problems with the Cougars mail system students may use their own private email system to send messages and attachments to the instructor.

All assignments (discussions and dropboxes) are due by midnight (central time) of the date due.

NO late Discussion Assignments will be received for credit.

Academic Honesty

All Columbia College policies are in effect as described in the Academic Dishonesty/Misconduct section of the current college Catalog. All your work must be your own unless collaboration has been authorized. If collaboration is authorized you must acknowledge the collaboration in writing. Your grade will be based in large part on the originality of your ideas and your written presentation of these ideas. Presenting as one's own the words, ideas, or expression of another in any form is cheating through plagiarism. If you are unsure what constitutes plagiarism, review the rules of original writing at the following web site: http://owl.english.purdue.edu/. This link provides valuable information, including examples about plagiarism. To review some plagiarism tools available to students, take a look at http://www.schoolsucks.com and www.termpapersites.com The content of these plagiarism sites would, if you were lucky, get you a "D" in this course if you were not caught. It is substandard work indeed, but you will almost always be caught if you try to cheat, due to the plagiarism prevention tools available to instructors. Here are two sites that may be of interest: http://www.indiana.edu/~istd and http://www.plagiarism.com Plagiarism will not be tolerated and the claim of ignorance is no excuse. Those found plagiarizing may fail the course.

Collaboration with other students is not permitted without explicit permission from the instructor. This is a form of plagiarism. Roommates and spouses taking the same course should be particularly careful.

Levels of Communication

We will be using a minimum of two levels of communication in this course, one formal, and the other informal. All email assignments are formal. They should be written as if you are communicating with a client. The formal rules of proper English and grammar apply for these submissions, and points will be deducted for misspellings, incomplete sentences, poor sentence structure, etc.

Discussion postings are informal. You do not have to use capitalizations to begin sentences; there are no penalties for misspellings, incomplete sentences, or other violations of grammatical rules. The criteria that have been met in conference postings are that your messages must be original and intelligible. You must communicate effectively. In addition, you must meet the weekly requirements for full credit on conference room assignments.
An optional communication tool we have at our disposal is the use of a chat room. Chat rooms allow us to communicate in a synchronous fashion if class participants desire to communicate with the instructor in "real time". If one or more students desire synchronous communication, indicate this in one of the discussion replies by requesting a chat session. We will establish the Chat option available in Desire2Learn, set a date and a time and discuss issues arising in the course. This course is structured around asynchronous communication.

Grading Policy

You will know in advance the standards for each assignment. My goal is to give you prompt, clear, and useful feedback to help you understand and program in the Java language.

You will be able to track your average exactly throughout the course. The grading scale is based on percentile as follows: A = 90-100; B = 80-89; C = 70-79; D = 60-69; F = 0-59. Each student is responsible for:

- Completing weekly reading assignments.
- Completing all weekly discussion questions in the online conference.
- Completing a total of 25 programming assignments, referred to as Dropbox assignments.
- Completing a Proctored Final Exam covering chapters 1 – 12 in the textbook.

Student Manual

There is a Student Manual on the Online Education website that will assist in helping to solve any problems you may have in the course. You are able to find the manual at:

- http://www.ccis.edu/online/studentmanual/

Proctoring Policy

You will have One (1) proctored exam. YOU MUST ARRANGE AN ACCEPTABLE PROCTOR. Acceptable proctors include but are not restricted to Columbia College faculty, Columbia College Extended Campus Directors or their staff, ministers, commanding officers (direct superiors are not allowed), corporate executive officers (again, direct supervisors are not allowed), and public librarians. I will be happy to entertain other suggestions. I must approve all proctors not on the CCEC list. If you are unable to locate a suitable proctor near you, check the list available through the NCTA Consortium of College Testing Centers.

- Columbia College Proctoring
    - http://www.ccis.edu/online/academics/proctor.asp

- NCTA Consortium of College Testing Centers
    - http://testing.byu.edu/NCTA/consortium/find.asp

There will probably be a fee to be tested at one of the College Testing Centers. I will need a name and email address for your proctor by the end of the second week of the term.

IV. Grades

Text readings should be completed prior to submitting assignments for the week.
Online discussion postings should be complete by Sunday of the assigned week. The discussion postings are worth 10 points each.

Dropbox Assignments will each be graded according to the point system presented in the assignment table below. Each Dropbox assignment will be worth 30 points.

Late assignments will not be accepted.

Exams: There will be one Proctored Final exam worth 200 points. The exam will consist of 10 programming problems worth 10 points each and 1 programming assignment worth 100 points. **This is a proctored open-book exam.**

WEEK 1 ASSIGNMENTS	PTS	DUE
Discussion Question 1 (Chapters 1-3)	10	Sunday
Chapter 1 – Programming Challenge 1, *Your First Java Program*, pages 28-29	30	Sunday
Chapter 2 – Programming Challenge 9, *Miles-per-Gallon*, page 109	30	Sunday
Chapter 3 – Programming Challenge 1, *Employee Class*, page 166	30	Sunday
WEEK 2 ASSIGNMENTS		
Discussion Question 2 (Chapters 4 and 5)	10	Sunday
Chapter 4 – Programming Challenge 4, *Software Sales*, page 247	30	Sunday
Chapter 5 – Programming Challenge 5, *Hotel Occupancy*, page 311	30	Sunday
WEEK 3 ASSIGNMENTS		
Discussion Question 3 (Chapters 6 and 7)	10	Sunday
Chapter 6 – Programming Challenge 5, *Month Class*, page 391	30	Sunday
Chapter 7 – Programming Challenge 2, *Payroll Class*, page 485	30	Sunday
WEEK 4 ASSIGNMENTS		
Discussion Question 4 (Chapter 8)	10	Sunday
Chapter 8 – Programming Challenge 1, *Backward String*, page 540	30	Sunday
Chapter 8 – Programming Challenge 3, *Sentence Capitalizer*, page 541	30	Sunday
WEEK 5 ASSIGNMENTS		
Discussion Question 5 (Chapter 9)	10	Sunday
Chapter 9 – Programming Challenge 5, *Course Grades*, page 616	30	Sunday
WEEK 6 ASSIGNMENTS		
Discussion Question 6 (Chapter 10)	10	Sunday
Chapter 10 – Programming Challenge 1, *TestScores Class*, page 671	30	Sunday
Chapter 10 – Programming Challenge 7, *File Encryption Filter*, page 672	30	Sunday
Chapter 10 – Programming Challenge 8, *File Decryption Filter*, page 672	30	Sunday

WEEK 7 ASSIGNMENTS		
Discussion Question 7 (Chapters 11 and 12)	10	Sunday
Chapter 12 – Programming Challenge 3, *Dorm and Meal Plan Calculator*, page 835	30	Sunday
WEEK 8 ASSIGNMENTS		
Discussion Question 8 (JDBC Handout)	10	Saturday
Database Challenge (Databases and Java Handout)	30	Saturday
Final Exam	200	Saturday

GRADE	CRITERIA FOR ONLINE TEXTBOOK QUESTION POSTINGS
0	Student failed to post the assignment.
1 – 7	Student's assignment met basic requirements, but lacked proper formatting, commenting, or was a duplication of another student's efforts only slightly changed.
8-10	Substantial, original contributions that further the work of the class. Well-placed comments referencing book. Code formatted correctly.
0	Assignment is late.

GRADE	CRITERIA FOR DROPBOX ASSIGNMENTS
30	Assignment is on time; complete, properly formatted, properly commented.
15 – 29	Assignment is on time; but incomplete, not formatted correctly, little or no comments
0	Assignment is late.

Grades/points for all assignments, exams and class discussion postings will be posted in the Desire2Learn "Grades" in order that students may keep up with their progress in the course. If you submitted your assignment(s) on time and see a (-) in the grade book, it may mean that I did not receive it, please contact me immediately.

V. Required Texts

- Starting Out With Java: Early Objects – with CD. Gaddis, Tony. 3rd Edition, 2008. Pearson. ISBN: 0321497686

Textbooks for the course may be ordered as follows:

- ONLINE: http://bookstore.mbsdirect.net/columbia.htm
- PHONE: 1-800-325-3252
- FAX: 1-800-499-0143

For additional information about the bookstore visit http://www.mbsbooks.com/direct

Students: Please note that the use of an eBook carries certain risks: information may be missing due to copyright restrictions, the book cannot be resold to MBS, and an eBook purchase cannot be refunded.

Buying Used Text books or new books from other vendors

You may use the above ISBN #'s to try to locate cheaper text books than can be purchased through MBS Bookstore. However, be warned ahead of time that normally the delivery time is longer than the 3-5 days guaranteed by MBS. Also note that if you do order books from sources other than MBS, you are on your own. The College has considerable leverage with MBS, and can check on orders and deliver status, and in some cases have the books shipped overnight air when necessary. We have no such leverage with any other publisher. If your books arrive late when ordering from MBS, we will extend the assignment deadlines, but cannot do so if ordered from any other publisher. Why take a chance just to save a few dollars?

VI. Course Schedule

Week 1

Readings

- Chapter 1 – Introduction to Computers and Java, pages 1-19
- Chapter 2 – Java Fundamentals, pages 31-102
- Chapter 3 – A First Look at Classes and Objects, pages 113-161

On-Line Discussion

- Discussion Question 1 (Chapters 1-3) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - print, println, Escape characters, Variables, String concatenation, Data types, Arithmetic operations, Math class, Casting, Constants, String class methods, Scanner, JOptionPane, Converting data types, Passing arguments, Constructors
- Starting point:

```
==========================================================================
import java.util.Scanner;
public class DiscussionOne {
	public static void main(String[] args) {
		//Added by Michael Jenkins
		//Constructor for all non-static methods in this class file.
		//dissOne can be called for any other methods added to this class.
		//No need to add another constructor.
		//Page 139 discusses constructors
		DiscussionOne dissOne = new DiscussionOne();
		//Example call to method displaying a personal greeting
		dissOne.displayWelcomeMessage();
		//End of code by Michael Jenkins
	}
	/**
	 * Added by Michael Jenkins
	 * This method asks user for input (name)
	 * Then displays a personalized welcome
	 * message to the user.
```

```
	 */
	public void displayWelcomeMessage() {
		//Scanner class used to get input from user.
		//Pages 86-89
		Scanner userInput = new Scanner(System.in);
		//print and println are used to send output to the screen.
		//Pages 37-42
		System.out.print("Please type your full name: ");
		String name = userInput.nextLine();
		System.out.print("Welcome, " + name);
		System.out.println(", to the first Discussion Question.");
	}
}
```
==========

Dropbox Assignments

- Chapter 1 – Programming Challenge 1, *Your First Java Program*, pages 28-29
- Chapter 2 – Programming Challenge 9, *Miles-per-Gallon*, page 109
- Chapter 3 – Programming Challenge 1, *Employee Class*, page 166

Week 2

Readings

- Chapter 4 – Decision Structures, pages 169-240
- Chapter 5 – Loops and Files, pages 251-304

On-Line Discussion

- Discussion Question 2 (Chapters 4 and 5) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - if Statement, if-else Statement, Logical operators, Nested if statements, Comparing String objects, Conditional Operators, switch statement, Random class, DecimalFormat class, Increments, Decrements, while loops, do-while loop, for loop, Nested loops, break, File input/output
- Starting point:

==========

```
public class DiscussionTwo {
	public static void main(String[] args) {
		//Added by Michael Jenkins
		//Constructor for all non-static methods in this class file.
		//dissOne can be called for any other methods added to this class.
		//No need to add another constructor.
		//Page 139 discusses constructors
		DiscussionTwo dissTwo = new DiscussionTwo();
		//Example call to method displaying odd numbers between 1 and 10
		dissTwo.showOddNumbers(10);
```

```
		//End of code by Michael Jenkins
	}
	/**
	 * Added by Michael Jenkins
	 * This displays odd numbers between 1 and the passed in number
	 */
	public void showOddNumbers(int myNum) {
		//For loop that loops through myNum times
		//Pages 266-274
		for(int x = 1;x < myNum; x++) {
			//if statement that checks the remainder
			//if the remainder is 1 then the number is odd
			//Pages 169-178
			if(x % 2 == 1) {
				System.out.println(x);
			}
		}
	}
}
```

==========

Dropbox Assignments

- Chapter 4 – Programming Challenge 4, *Software Sales*, page 247
- Chapter 5 – Programming Challenge 5, *Hotel Occupancy*, page 311

Week 3

Readings

- Chapter 6 – A Second Look at Classes and Objects, pages 315-383
- Chapter 7 – Arrays and the ArrayList Class, pages 395-479

On-Line Discussion

- Discussion Question 3 (Chapters 6 and 7) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - Static Class, Overloading Methods, Overloading Constructors, toString, Passing Objects as arguments, Aggregation, this reference variable, Inner Classes, Enumerated Types, Equals, Array, ArrayList, Two-dimensional Arrays, Arrays of Objects
- Starting point:

==========

```
public class DiscussionThree {
	public static void main(String[] args) {
		//Added by Michael Jenkins
		//Example call to method displaying odd numbers between 1 and 10
		//While passing in an int Array
		//Chapter 7
```

```
                int[] myNum = {1,2,3,4,5,6,7,8,9,10};
                showOddNumbers(myNum);
                //End of code by Michael Jenkins
        }
        /**
         * Added by Michael Jenkins
         * This displays odd numbers between 1 and the passed in number
        */
        public static void showOddNumbers(int[] myNum) {
                //For loop that loops through myNum times
                //Pages 266-274
                for(int x = 0;x < myNum.length; x++) {
                        //if statement that checks the remainder
                        //if the remainder is 1 then the number is odd
                        //Pages multiple in Chapter 7
                        if(myNum[x] % 2 == 1) {
                                System.out.println(myNum[x]);
                        }
                }
        }
}
```

==========

Dropbox Assignments

- Chapter 6 – Programming Challenge 5, *Month Class*, page 391
- Chapter 7 – Programming Challenge 2, *Payroll Class*, page 485

Week 4

Readings

- Chapter 8 – Text Processing and Wrapper Classes, pages 491-536

On-Line Discussion

- Discussion Question 4 (Chapter 8) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - Wrapper Classes, Character class, String Objects (many opportunities here), StringBuilder class, Tokenizing Strings
- Starting point:

==========

```
public class DiscussionFour {
        public static void main(String[] args) {
                //Added by Michael Jenkins
                //Example call to a static method converting text to upper/lower case
                conversionTest("Michael Jenkins Teaches CISS238!");
                //End of code by Michael Jenkins
        }
        /**
```

```
         * Added by Michael Jenkins
         * This displays message in upper/lower case
        */
        public static void conversionTest(String value) {
                //converts value to upper/lower case
                //page 497
                System.out.println("Original: " + value);
                System.out.println("UPPER: " + value.toUpperCase());
                System.out.println("lower: " + value.toLowerCase());
        }
}
```

==========

Dropbox Assignments

- Chapter 8 – Programming Challenge 1, *Backward String*, page 540
- Chapter 8 – Programming Challenge 3, *Sentence Capitalizer*, page 541

Week 5

Readings

- Chapter 9 – Inheritance, pages 545-608

On-Line Discussion

- Discussion Question 5 (Chapter 9) – In this discussion question, you are given the beginnings of an application. Two classes have been created for you. Using the knowledge you gained from this week's chapters, add a method or methods, class or classes that is different from any other student's submission (as well as the example). Use some of the suggestions below in your code. Ensure that you comment your code providing a description of what your code does as well as references to where your use of code can be found in the textbook.
    - Superclass Constructor, Superclass Methods, Protected Members, Inheritance, Polymorphism, Abstract Classes, Abstract Methods
- Starting point:

==========

```
public class DiscussionFive extends Shape {
        private double radius;
        public void setRadius(double r) {
                radius = r;
                setArea(Math.PI * r * r);
        }
        public double getRadius() {
                return radius;
        }
}
public class Shape {
     private double area;
     public void setArea(double a) {
          area = a;
     }
     public double getArea() {
          return area;
     }
}
```

==========

Dropbox Assignments

- Chapter 9 – Programming Challenge 5, *Course Grades*, page 616

Week 6

Readings

- Chapter 10 – Exceptions and Advanced File I/O, pages 619-644

On-Line Discussion

- Discussion Question 6 (Chapter 10) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - Exception handling, Throwing Exceptions, File input/output
- Starting point:

==========

```
public class DiscussionSix {
	public static void main(String[] args) {
		//Added by Michael Jenkins
		//Example for exception handling
		//successful
		convertToInt("1");
		//thows exception
		convertToInt("H");
		//End of code by Michael Jenkins
	}
	/**
	 * Added by Michael Jenkins
	 * This catches an exception
	 */
	public static void convertToInt(String value) {
		//converts value to int
		//page 634
		try{
			System.out.println("Converted ok: " + Integer.parseInt(value));
		}catch(NumberFormatException nfe){
			System.out.println("That's not a number.");
		}
	}
}
```

==========

Dropbox Assignments

- Chapter 10 – Programming Challenge 1, *TestScores Class*, page 671

- Chapter 10 – Programming Challenge 7, *File Encryption Filter*, page 672
- Chapter 10 – Programming Challenge 8, *File Decryption Filter*, page 672

Week 7

Readings

- Chapter 11 – GUI Applications – Part I, pages 673-764
- Chapter 12 – GUI Applications – Part II, pages 773-829

On-Line Discussion

- Discussion Question 7 (Chapters 11 and 12) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - Dialog boxes, Windows, Borders, JPanel, Splash Screens, Radio Buttons, Check Boxes, Text fields, Lists, Combo Boxes, Images, Labels, Buttons, File Choosers, Color Choosers, Menus, Sliders
- Starting point:

==========

```
import javax.swing.*;
public class DiscussionSeven {
	public static void main(String[] args) {
		//Added by Michael Jenkins
		getAverage();
	}
	/**
	 * Added by Michael Jenkins
	 * from the book on page 684
	 * This is just a starting point.
	 */
	public static void getAverage() {
		int score1, score2, score3; // Three test scores
		String strInput; // String input
		double average; // Average test score
		int repeat; // Confirm dialog button clicked
		do {
			// Get the first test score.
			strInput = JOptionPane.showInputDialog(null, "Enter score #1.");
			score1 = Integer.parseInt(strInput);

			// Get the second test score.
			strInput = JOptionPane.showInputDialog(null, "Enter score #2.");
			score2 = Integer.parseInt(strInput);

			// Get the third test score.
			strInput = JOptionPane.showInputDialog(null, "Enter score #3.");
			score3 = Integer.parseInt(strInput);

			// Calculate and display the average test score.
			average = (score1 + score2 + score3) / 3.0;
			JOptionPane.showMessageDialog(null, "The average is " + average);
```

```
                        // Does the user want to average another set?
                        repeat = JOptionPane.showConfirmDialog(null,"Would you like to average
another set of test scores?", "Please Confirm.",JOptionPane.YES_NO_OPTION);
                } while (repeat == JOptionPane.YES_OPTION);
                System.exit(0);
        }
}
```
==========

Dropbox Assignments

- Chapter 12 – Programming Challenge 3, *Dorm and Meal Plan Calculator*, page 835

Week 8

Readings

- Databases and Java Handout

On-Line Discussion

- Discussion Question 8 (Chapter 8) – In this discussion question, you are given the beginnings of an application. The main method has been created for you as well as an example method. Using the knowledge you gained from this week's chapters, add a method that is different from any other student's submission (as well as the example) and add a line to the main method to test your newly created method. Use some of the suggestions below in your method. Ensure that you comment your code providing a description of what your method does as well as references to where your use of code can be found in the textbook.
    - JDBC Connection, executeUpdate, executeQuery, preparedStatement, Statement, SELECT, UPDATE
- Starting point:

==========

```
import java.sql.*;
public class DiscussionEight {
        public static void main(String[] args) {
                //Added by Michael Jenkins
                //Method to run a database test
                testDBConnect("c:/testdb.mdb");
        }
        /**
         * Method to test a database connection and pull back some data
         * Added by Michael Jenkins
         * Supplement
         * @param fileLocAndName
         */
        public static void testDBConnect(String fileLocAndName) {
                try {
                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                        // set this to a MS Access DB you have on your machine
                        String database = "jdbc:odbc:Driver"
                                + "={Microsoft Access Driver (*.mdb)};DBQ="
                                + fileLocAndName + ";DriverID=22;READONLY=true}";
                        // get the connection from the DriverManager
```

```
                        Connection conn = DriverManager.getConnection(database, "", "");
                        Statement stmt = conn.createStatement();
                        stmt.execute("select FirstName, LastName, HomePhone from PhoneList");
                        //get any ResultSet that came from the query
                        ResultSet rs = stmt.getResultSet();
                        if (rs != null) {
                                // this will step through the data row-by-row
                                while (rs.next()) {
                                        // Output to the screen.
                                        // rs.getString(1) will get the first column from the query
                                        System.out.println(rs.getString(1) + " "
                                                        + rs.getString(2) + ", "
                                                        + rs.getString(3));
                                }
                        }
                        stmt.close();
                        conn.close();
                } catch (Exception e) {
                        System.out.println("Error: " + e.getMessage());
                }
        }
}
================================================================================
```

Dropbox Assignments

- Databases and Java Handout

Exam

The proctored Final will again be a computer proctored test in the testing section of Desire2Learn. Your proctor will be given the password. If you are near a local Columbia College campus, go to the proctoring web page and coordinate with a proctor.

- http://www.ccis.edu/DistanceEducation/proctoring.html

Inform me during week 2 of the name and the email address of the proctor so I may send them instructions for the Final. The exam will consist of 10 programming problems worth 10 points each and 1 programming assignment worth 100 points. You will have 2 hours to complete the exam. The exam is open book.

You may take the Final any time Monday thru Saturday coordinated with your proctor. Please make sure you coordinate to complete the exam No Later than 5:00 pm CST so I will have the results before close of business on Saturday.

VII. Instructor Information

Michael Jenkins

Work:	720-842-6370
Cell:	303-378-8513
Email Address:	msjenkins@cougars.ccis.edu
Location:	Aurora, Colorado