AMERICAN UNIVERSITY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

CSI 311 – Java Programming

Section B

Spring Term 2011-2012
Ashrafieh Campus

INSTRUCTOR	Michel Owayjan, M.E. mowayjan@aust.edu.lb	
LECTURE HOURS	TTH	02:00 pm – 03:15 pm
OFFICE HOURS	WF	02:00 pm – 03:00 pm
	T-TH	03:30 pm – 04:30 pm
	Otherwise by appointment	
	Room:	805, Block A

COURSE OBJECTIVES

This course studies the aspects and types of the Java programming language. It explores the differences among Java Scripts, Java Applets, and Java stand alone Applications. Topics include: A discussion of JavaScript – The de facto client side scripting language for web-based applications, the programming foundation for client side scripting. Java Applets to create programs that can be executed simply by loading the appropriate web page in a web browser. The basic constructs of the Java language which include access controls, flow control, object orientation, inheritance, polymorphism, exception handling, garbage collection, threads, and sockets to create stand alone applications that can be executed using the Java Interpreter.

COURSE PREREQUISITES

CSI 205 Computer Programming I

PREREQUISITES BY TOPICS

The student should have a background in computers from the point of view components, both hardware and software. Also he should have a background on the basics of any high level programming language. Familiarity with Object Oriented programming language is suggested, but not required.

COURSE CREDITS 3 Credit Hours

INSTRUCTION TECHNIQUE

Lecture will be used predominantly. These will be supported by handouts and problem sets.

REQUIRED TEXTS

- Java How to Program by Paul Deitel & Harvey Deitel, Ninth Edition, Pearson

REFERENCES

- Problem Solving with Java, by Koffman, Wolz, Second Edition
- Introduction to java programming, by Daniel Liang, Eighth Edition, Pearson
- Java for Students by Bell, Parr, 3rd Edition
- The Object of Java by David D. Riley

OTHER REQUIREMENTS

- JAVA 2 Platform, JDK 1.6.0® (Sun-Oracle, Inc 2010). Standard Edition

GRADE DISTRIBUTION

This course involves a number of activities, ranging from lectures, problem sets, quizzes, projects, and exams, to the midterm and final exams. All of these attributes of the course are intended to help the student in developing his/her understanding of the material covered in CSI 311 and in providing the department and the course instructor with information on how the student is doing. Consequently, all of these activities are considered to be vital and will be taken into consideration while assigning grades at the end of the term. When the student is being assigned a course grade, it is imperative that this grade accurately reflects the student’s level of achievement and his/her mastery of the material covered in CSI 311. A breakdown of the weighting that will be used in making this assessment is as follows:

Lab Work (LW)	20%
Class Attendance (ATT)	05%
Quizzes / Problem Sets (QZ)	10%
Projects (PRJ)	10%
Exam I (EX I)	10%
Midterm Exam (MID)	15%
Exam II (EX II)	10%
Final Exam (FIN)	20%

The following scale will be used to assign course letter grades:

90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
BELOW 60	F

Note: *Make-up exams and late assignments will be dealt with according to the distributed course rules and regulations, which are governed by the Department of Computer Science.*

However, students have to realize that in the event that a make-up exam is granted, it would be unjust that the grantee becomes privileged over his/her fellow colleagues in terms of extra-time to prepare for the exam and in developing an idea about the contents of the exam. Accordingly, the make-up exam will carry an increased level of difficulty of at least 20% from the regular exam.

COURSE COMPETENCIES

- Introduce students to Object Oriented Systems Development and design.
- To understand and apply Object-Oriented programming using Java language.
- Build teamwork proficiency and negotiating skills in the software development process.

ATTENDANCE

For legitimate reasons only, a student is allowed to absent him/herself for a maximum of **4 contact hours from the course lectures sessions**. However, any absence of more than two sessions will be counted toward the 5% evaluation of the course grade. Absences beyond the specified maximum limit will result in an automatic **AW** in the course and possibly other disciplinary measures. An unexcused absence from a quiz will result in a zero on that quiz. Should the student be absent from a lecture during which a *problem set* or a *laboratory assignment* is due, it is the duty of the student to make certain that the homework assignment is handed in on time. Failure to do so will incur a penalty on that assignment's grade.
Note: 7.5 points will be deducted from the class attendance grade for every non excused absence.

ACADEMIC INTEGRITY

The student should get familiar with the ***Course Rules and Regulations*** of the Department of Computer Science at AUST. These are compiled and distributed to the student at the beginning of every academic term. In particular, the student should be aware that plagiarism, abuse of laboratory facilities, and other sorts of academic dishonesty are not tolerated and can result in unsympathetic penalties.

The Department of Computer Science (CSI) fully acknowledges the potential significance of students studying together. In this sense, the CSI Department does not have any reservation to this kind of collaboration, as long as all contestants are involved in all facets of the work, and not with each individual contributing to a fraction of the assignment. Specifically, when a student submits an assignment with his/her name on it, the CSI Department takes it for granted that the details presented in the assignment are entirely the student's own work, and that this student has substantially participated in the creation of this work. If a portion of the work has been conceived by collaborative work, that section should be highlighted and the names of the students involved in this collaboration should be listed next to that section.

All projects and problem sets are expected to be handed in at the beginning of the lecture of the due date. Late project/problem sets are permissible if the solution is not published yet, but will be penalized. If a project or problem set is not submitted at the beginning of the lecture of the due date, it will automatically be considered as late. In accordance, the project/assignment will loose 25% of its merit. However, the student has the right to hold on to it and submit it at the beginning of the very next lecture. If this situation is repeated, another 25% of the total merit of the project/assignment will be deducted until the project/assignment looses all of its merit.

PROBLEM SETS

For most part of the course, *problem sets* will be assigned on a weekly basis. A detailed description of the *problem set* assignment timetable is compiled at the end of this handout.

COURSE OUTLINE BY TOPIC

The day-to-day topics to be covered in the table below may be adjusted as the subject proceeds, but all examination dates are fixed, and problem set due dates are unlikely to change.

Course Outline

	Week No	*Day*	Date (DD/MM/YYYY)
Chapter 1: Introduction to Computers and Java	**1**		
- Introduction.		T	21/02/2012
- Why java (Explain the characteristics of java)		T	21/02/2012
- The Java Environment and Compiler. Installing the J2SE Development Kit.		T	21/02/2012
Chapter 2: Introduction to Java Applications		T	21/02/2012
- Our First Program in Java: Printing a Line of text		T	21/02/2012
- Modifying Our First program		T	21/02/2012
- Displaying Text with printf		Th	23/02/2012
- Another Application: Adding Integers.		Th	23/02/2012
- Memory Concepts, Arithmetic.		Th	23/02/2012
- Decision Making: Equality and Relational Operators.		Th	23/02/2012
Problem Solving (Ex 2.28 and 2.30 pages 104-105 Live demo)		Th	23/02/2012
Chapter 3: Introduction to Classes and Objects, Methods and Strings	**2**	**T**	**28/02/2012**
- Classes, Objects, Methods and Instance Variables.		T	28/02/2012
- Declaring a Class with a Method and Instantiating an Object of a class.		T	28/02/2012
- Declaring a Method with a Parameter.		T	28/02/2012
- Instance Variables, set Methods and get Methods.		T	28/02/2012
- Primitive Types vs. Reference Types.		T	28/02/2012
- Initializing Objects with Constructors.		T	28/02/2012
- Floating-Point Numbers and Type double.		T	28/02/2012
- GUI and Graphics Case Study: Using Dialog Boxes.		T	28/02/2012
Problem Solving (Ex 3.5 page 104 Live demo)		**T**	**28/02/2012**
Chapter 4: Control Statements: Part I			
- Control Structures.		Th	01/03/2012
- Selection statements.		Th	01/03/2012
- while Repetition Statement.		Th	01/03/2012
- Compound Assignment Operators.		Th	01/03/2012
- Increment and Decrement Operators. Primitive Types.		Th	01/03/2012
- GUI and Graphics Case Study: Creating Simple Drawings		Th	01/03/2012
Chapter 5: Control Statements Part 2			
- for Repetition Statement.			
- Section 5.5: do…while Repetition Statement.		Th	01/03/2012
- switch Multiple-Selection Statement.		Th	01/03/2012

	Week	***Day***	**Date (DD/MM/YYYY)**
- break and continue Statements.		Th	01/03/2012
- Logical Operators.		Th	01/03/2012
- GUI and Graphics Case Study: Drawing Rectangles and Ovals.		Th	01/03/2012
Problem Solving (Live demo of GUI and Graphics Case Studies, Creating simple Drawing, pages 138-140 and Drawing Rectangles and Ovals pages 189-191))	**3**	**T**	**06/03/2012**
Chapter 6: Methods: A Deeper Look		T	06/03/2012
- static Methods, static Fields and Class Math.		T	06/03/2012
- Declaring Methods with Multiple Parameters.		T	06/03/2012
- Argument Promotion and Casting.		T	06/03/2012
- Java API Packages.		Th	08/03/2012
- Case Study. Random-Number Generation		Th	08/03/2012
- Scope of Declarations.		Th	08/03/2012
- Method Overloading.		Th	08/03/2012
- GUI and Graphics Case Study: Colors.		Th	08/03/2012
Problem Solving (Live demo of the Case Study: A game of Chance; Introducing Enumerations) and live demo of Ex 6.32 page 245.	**4**	**T**	**13/03/2012**
Chapter 7: Arrays and ArrayLists		T	13/03/2012
		T	13/03/2012
- Declaring and Creating Arrays.		T	13/03/2012
- Examples Using Arrays.		T	13/03/2012
- Case Study.		Th	15/03/2012
- Enhanced for Statement.		Th	15/03/2012
- Passing Arrays to Methods.		Th	15/03/2012
- Case Study: Using an Array to Store Grades.		Th	15/03/2012
- Multidimensional Arrays.		Th	15/03/2012
- Case Study: Using a Two Dimensional Array.		Th	15/03/2012
- Variable-Length Argument Lists.		Th	15/03/2012
- Using Command-Line Arguments.		Th	15/03/2012
- Class Arrays	**5**	T	20/03/2012
- Introduction to Collections and Class ArrayList		T	20/03/2012
Problem Solving (Live demo, Create a java application using the methods: fillarray() that fills an array of integers and or Strings, sortarray() that sorts the array, swaparray() that swaps the array elements, and then printarray() that prints the array using both printf method and JOptionPane class.)		**T**	**20/03/2012**
EXAM I		**Th**	**22/03/2012**
Chapter 8: Classes and Objects: A Deeper Look			
- Introduction.	**6**	T	27/03/2012
- Time Class Case Study.		T	27/03/2012
- Controlling Access to Members.		T	27/03/2012
- Referring to the Current Object's Members with the ***this*** Reference.		T	27/03/2012
- Time Class Case Study: Overloaded Constructors.		T	27/03/2012
- Default and No-Argument Constructors.		T	27/03/2012
- Notes on Set and Get Methods.		T	27/03/2012
- Composition.		T	27/03/2012
		Th	29/03/2012
	Week No	***Day***	**Date (DD/MM/YYYY)**

- Enumerations.	**7**		
- Garbage Collection and Method finalize.		Th	29/03/2012
- static Class Members.		Th	29/03/2012
- static Import. final Instance Variables.		Th	29/03/2012
		Th	29/03/2012
- Time Class Case Study: Creating Packages.		Th	29/03/2012
- Package Access.		Th	29/03/2012
Problem Solving (Live Demo, Create only one complete Student application demonstrating <u>all</u> topics covered in Chapter 8) and (Live demo of Ex 8.15 page 393)		**T**	**03/04/2012**
Chapter 9: Object-Oriented Programming: Inheritance		T	03/04/2012
- Introduction.		T	03/04/2012
- Superclasses and Subclasses.		T	03/04/2012
- protected Members.		T	03/04/2012
- Relationship between Superclasses and Subclasses.	**8**	Th	05/04/2012
- Constructors in Subclasses.		Th	05/04/2012
- Object Class.		Th	05/04/2012
- GUI and Graphics Case Study.		Th	05/04/2012
Chapter 10: Object-Oriented Programming: Polymorphism		T	10/04/2012
- Introduction.		T	10/04/2012
- Polymorphism Examples.		T	10/04/2012
- Demonstrating Polymorphic Behavior.		T	10/04/2012
- Abstract Classes and Methods.		Th	12/04/2012
- **Live Demo of Case Study: Payroll System Using Polymorphism**.		Th	12/04/2012
Revision		**T**	**17/04/2012**
MIDTERM	**9**	**Th**	**19/04/2012**
	10	**F**	**27/04/2012**
<u>Final Project Distribution</u>		Th	03/05/2012
- **Final Methods and Classes.**		Th	03/05/2012
- **Case Study: Creating and Using Interfaces**		Th	03/05/2012
		Th	03/05/2012
Chapter 14: GUI Components Part I		Th	03/05/2012
- Introduction.		Th	03/05/2012
- Java's New Numbus Look-and-Feel	**11**	T	08/05/2012
- Simple GUI-Based Input/Output with JOptionPane.		T	08/05/2012
- Overview of Swing Components.		T	08/05/2012
- Displaying Text and Images in a Window.		T	08/05/2012
- Text Fields. Event Handling with Nested Classes.		T	08/05/2012
- Common GUI Event types and Listener Interfaces.		T	08/05/2012
- How Event Handling Works.		T	08/05/2012
- JButton		T	08/05/2012
- Buttons That Maintain State.		T	08/05/2012
		Th	10/05/2012
- JComboBox and Using an Anonymous Inner Class for Event Handling		Th	10/05/2012
		Th	10/05/2012
- JList/Multiple-Selection Lists		Th	10/05/2012
- Mouse Event Handling		Th	10/05/2012
- Adapter Classes		Th	10/05/2012
- Jpanel Subclass for Drawing with the Mouse		Th	10/05/2012
- Key Event Handling		Th	10/05/2012

	Week No	***Day***	**Date (DD/MM/YYYY)**
- Introduction to Layout Managers.		Th	10/05/2012
- Using Panels to Manage More Complex Layouts.		Th	10/05/2012
- JTextArea.		**Th**	10/05/2012
		Th	10/05/2012
Problem Solving Live demo Ex. 14.12 and 14.13 page 663		**T**	**15/05/2012**
	12		
EXAM II			
Chapter 23: Applets and Web Start	**13**	Th	17/05/2012
- Introduction.		Th	17/05/2012
- Sample Applets Provided with JDK.		Th	17/05/2012
- Simple Java Applet: Drawing a String.		Th	17/05/2012
- Applet Life-Cycle Methods.		Th	17/05/2012
- Initializing an Instance Variable with Method **init**.		Th	17/05/2012
- Sandbox Security Model.		Th	17/05/2012
- Java Web Start and the Java Network Launch Protocol (JNLP)		Th	17/05/2012
Problem Solving Live Demo of a previously written JFrame application and convert it to JApplet	**14-15**	T	21/05/2012
Chapter 11: Exception Handling: Deeper Look		T	22/05/2012
- Introduction.		T	22/05/2012
- Exception-Handling Overview.		T	22/05/2012
- Divide By Zero Without Exception Handling.		T	22/05/2012
- Example: Handling ArithmeticExceptions and InputMismatchExceptions. **(Live Demo)**		T	22/05/2012
		Th	24/05/2012
- When to Use Exception Handling.		Th	24/05/2012
- Java Exception Hierarchy.		Th	24/05/2012
- finally block.		Th	24/05/2012
- Stack Unwinding.		Th	24/05/2012
- printStackTrace, getStackTrace and getMessage.		Th	24/05/2012
- Chained Exceptions.		Th	24/05/2012
- Declaring New Exception Types.		Th	24/05/2012
- Preconditions and Postconditions.		Th	24/05/2012
- Assertions.		Th	24/05/2012
		Th	24/05/2012
Chapter 26: Multithreading	**16**	T	29/05/2012
- Introduction.		T	29/05/2012
- Thread States: Life Cycle of a Thread.		T	29/05/2012
- Thread Priorities and Thread Scheduling.		Th	31/05/2012
- Creating and Executing Threads Using the Thread class and the		Th	31/05/2012
Runnable Interface		Th	31/05/2012
- Thread Synchronization.		Th	31/05/2012
Collecting Final Projects and students' presentation		T	05/06/2012
	17		
READING PERIOD		TH	07/06/2012
		F	08/06/2012
FINAL EXAM WEEK	**18**	**S**	09/06/2012
		M	18/06/2012

PROBLEM SETS WITH DUE DATES

Problem Set No. 1: *Read self-review and solve 2.8, 2.9, 2.10, 2.11, 2.12, 2.31 and 2.33 pp: 102-105 (Due Thu. 23/02/2012).*
Problem Set No. 2*: Read self-review and solve 3.5, 3.6, 3.7, 3.11, 3.13 pp. 136 (Due Thu. 01/03/2012).*
Problem Set No. 3: *Read self-review and solve 4.16, 4.17, 4.22 pp. 181-183 (Due Tue. 06/03/2012).*
Problem Set No. 4: *Read self-review and solve 5.13, 5.17, 5.19, 5.24 pp. 229-231 (Due. Thu. 08/03/2012).*
Problem Set No. 5: *Read self-review and solve 6.7, 6.8, 6.9, 6.11, 6.12, 6.13, 6.20, 6.22 pp 270 – 273 (Due Tue. 13/03/2012).*
Problem Set No. 6: *Read self-review and solve 7.9, 7.10, 7.17 and 7.19 pp. 331 – 333 (Due Thu. 22/03/2012).*
Problem Set No. 7: *Read self-review and solve 8.4, 8.6, 8.11, 816 pp. 391 – 393 (Due Tue. 03/04/2012).*
Problem Set No. 8: *Read self-review and solve 10.8, 10.9, 10.10 pp. 472 (Due Tue. 17/04/2012).*
Problem Set No. 9: *Read self-review and solve 14.5, 14.6, 14.7, 14.8, 14.9, 14.11, 4.13 pp. 661-663 (Due Thu 17/05/2012).*
Problem Set No. 11: *Read self-review and solve 11.15, 11.17, 11.19, 11.20 pp.504 (Due Thu. 31/05/2012).*

Correction of the homework will be delivered the following week

DISCLAIMER: Changes may be performed to the above syllabus without any prior notification.

Copyright ©2011-2012 American University of Science & Technology - Department of Computer Science. All rights reserved.