ISM 515: Object Oriented Programming
Naveed Baqir
Spring 2009

Course Syllabus

Office Location: 479 Bryan Building
Office Phone: 336-256-0192
Cell: 336-375-4138
Office Fax: 336-334-4083
E-Mail: mnbaqir@spartan.uncg.edu
Office Hours: W 4:00 PM to 6:00 PM, Other times by appointment

Catalog Description
This course is aimed at students who wish to learn how to develop applications in Java. This course will also provide an overview of Object Oriented Programming concepts using Java.

Rational for ISM 515:
Internet has fundamentally changed all aspects of conducting business. One of the biggest technological factors that organizations have to overcome when attempting to utilize the Internet to conduct business is how to deliver the necessary application code that will be used to access and manipulate the information over the Internet. One of the technologies that help to overcome the application code distribution to end-users is Java. Java is a programming language that was introduced by Sun Microsystems and was originally designed for use in consumer electronic devices and to be platform independent. Java has since been adapted for the distributed computing environment that includes the Internet. Java programs can run on client workstations and servers. With recent developments, Java has become not only the programming language for developing applets that are typically found as small applications that are embedded within Web pages, but also for developing full fledged applications. Using Java, programmers can write sophisticated programs that can either run as an application or can be distributed through the Web to just about any computer in the world thus making Web pages more interactive.

ISM 515 Coverage:
ISM 515 aimed at students who wish to learn how to program in Java. Basic Object Oriented Programming (OOP) concepts, including objects, classes, methods, parameter passing, information hiding, inheritance and polymorphism are introduced and their implementations using Java are discussed. Java language elements and characteristics, including data types, operators, control structures are discussed in order to develop Java applications. The course also intended for students who would like to learn how to develop internet based applications and applets in Java. This course provides an overview of characteristics of Java applets and discusses their development. The course also covers a variety of other advanced topics, including developing graphical user interface (GUI), and graphics in both AWT and SWING. Depending on the interest of the students and time permitting, a number of more advanced Java topics will also be discussed. These may include: Creating Java database connectivity with JDBC; Manipulating databases with JDBC; Programming for Internet and World Wide Web; Servlet’s overview and architecture; Handling HTTP get requests containing data and handling HTTP post requests; redirecting requests to other resources; Multi-Tier applications; Using JDBC from a Servlet; and an overview of JavaServer pages.

ISM 515 Objectives:
- To understand Object Oriented Programming concepts
- To understand the role that methods play in an object-oriented program
To understand the concept of a class hierarchy.
To know the basic characteristics of Java
To become familiar with the relationship between classes and objects in a Java program
To comprehend Java Data and Control Structures
To understand the difference between a Java application and a Java applet.
To understand the role that data plays in effective program design
To be able to solve problems involving repetition
To understand the goals and principles of structured programming
To write efficient and effective applications in Java
Be able to design and implement a Java Applet.
Understand Java's event handling model.
Be able to handle button clicks in your programs.
Have a better appreciation of inheritance and polymorphism.
Know how to design a simple graphical user interface (GUI).
Acquire a familiarity with the Swing component set.
Understand the relationship between the AWT and Swing.
Have a better understanding of Java's event model.
Be able to design and build simple Graphical User Interfaces (GUI)s.
Appreciate how object-oriented design principles were used to extend Java's GUI capabilities.
Additional topics depending on time and the interests of the students may include:
  • Networking
  • Java Database Connectivity with JDBC™.
  • Servlets.
  • JavaServer Pages (JSP)

Required Textbook:
Introduction to Java Programming, By: Y. Daniel Liang. (I am not sure which one the book store has. Even if you are buying it online, please make sure to check it with the UNCG book store about the Edition information for this book)
Other supplemental books as needed.
Please bring your Textbook to every class meeting

Instructional Methods:
To learn Java, like any other programming language, students must practice code writing and code analyzing. To enhance learning both elementary and advanced Java issues, techniques and concepts covered in this course, we will work on writing and analyzing hundreds of small, well-annotated example programs and/or program segments. To be able to analyze and critique different approaches to similar problems, the class requires students to survey other students’ coding attempts in detail. To convey the bulk of the knowledge required to succeed as a programmer and a developer, we will rely on the textbooks, class lectures, class discussions, in-class exercises, and lab exercises. To further facilitate learning, each lesson is complemented with quizzes, exercises, other Q&A; students are required to read and learn from technical materials in order to use Java. Homework assignments and class projects challenge the student to apply the knowledge learned in class.

Tests:
There are two exams – a mid term exam and a final. Parts of tests may require the use of a computer or maybe take home. Attendance on the date of tests is MANDATORY. No make up examinations are offered for any reason. For each exam, I will assign which chapters and which questions you should answer. I will give you further instructions in class.
**Programming Assignments:**

Based on my experience teaching programming, the ONLY way to learn a programming language is to write programs in that language. You MUST attempt to solve many different problems. You MUST examine and implement many different programs. You MUST see different ways of doing the same thing to compare them and learn the best computational way. You MUST FAIL to SUCCEED!! The only way to achieve all this is by practice writing programs. Therefore, programming assignments and projects constitute a major portion of the requirements for this course. You are required to complete a number of programming assignments. More details will be forthcoming during the course of the semester. Each programming assignment is due at the beginning of the class on the scheduled due date. We will spend a few minutes discussing the concepts covered by the assignment and discuss different implementations of it. Students may not skip class to work on programming assignments. Excuses about the network, printers, or overcrowding in the labs, etc., will not be accepted. Please do not procrastinate on the programming assignments. Because of the level of complexity and detail required to learn any programming language, especially Java, you should expect to spend no less than 10 to 20 hours per week reading the material and working on exercises and assignments. The practice of meeting commitments is a fundamental obligation with meaningful consequences in a real work setting. There are no extra credit assignments.

**Attendance Policy:**

Each student is responsible for all the information and announcements that are made in class. Poor performance in this course is directly related to poor attendance. Students are expected to conduct themselves in a professional manner. A professional shows up for scheduled meetings prepared and on time. Any student missing the first two classes without notifying me will be administratively dropped from the course. Any student missing more than three classes (excused or not) may have their grade dropped by a letter grade.

**Grading:**

Grades for the course are based on tests and lab assignments. The course grade will be calculated using the following weights:

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Percentage of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Assignments</td>
<td>45%</td>
</tr>
<tr>
<td>Exams (1 @ 20%, 1 @ 30%)</td>
<td>50%</td>
</tr>
<tr>
<td>Attendance and Instructor’s Discretion</td>
<td>5%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>

The final grade is determined using the following 10-point scale:

- 970-1000 : A+  
- 930-969 : A  
- 900-929 : A-  
- 870-899 : B+  
- 830-869 : B  
- 800-829 : B-  
- 730-799 : C  
- 700-729 : C-  
- 650-699 : D  
- Below 650 : F

**Ethical Issues and the Honor Code Policies:**

University students are expected to conduct themselves in accordance with the highest standards of academic honesty. Academic misconduct for which a student is subject to penalty includes all forms of cheating, such illicit possession of examinations or examination materials, forgery, or plagiarism. Students will NOT make, borrow, or "share" copies of their lab assignments or files with other students. Plagiarism is defined as "presenting as one’s own work that work which is, in whole or in part, the work of another person or persons without giving proper credit to the appropriate source." This includes submitting work done by another, as one’s own work. It is understood that what you turn in to me for grade represents your own effort. Plagiarism will be immediately punished with a grade of zero for the assignment in question. Further disciplinary action will be pursued as I deem appropriate. Helping one another is allowed, but copying is cheating. This practice is against the UNCG Honor Code and defeats the purpose of this course. No credit will be received for shared work, and other penalties may be imposed. I will pursue cheating as far as the university allows me.
**Attendance Policy:**
Each student is responsible for all the information and announcements that are made in class. Poor performance in this course is directly related to poor attendance. Students are expected to conduct themselves in a professional manner. A professional shows up for scheduled meetings prepared and on time. Any student missing the first two classes without notifying me will be administratively dropped from the course. Any student missing more than three classes (excused or not) may have their grade dropped by a letter grade.

**Tentative Course Outline**
(Please note: This course outline is a tentative outline, we will make every attempt to follow this outline. However, deviations from this outline may become necessary)

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| 1    | Jan 21 | Chapter 1 Introduction to Computers, Programming, and Java  
Chapter 2 Primitive Data Types and Operations  
Lab on Java IDE: Dr. Hamid Nemati will cover this class | |
| 2    | Jan 28 | Chapter 3 Selection Statements | Assignment 1 Due |
| 3    | Feb 4  | Chapter 4 Loops  
Chapter 5 Methods | Assignment 2 Due |
| 4    | Feb 11 | Chapter 6 Arrays  
Chapter 8 Strings and Text I/O  
Chapter 17 Exceptions and Assertions | Assignment 3 Due |
| 5    | Feb 18 | Chapter 7 Objects and Classes  
Chapter 9 Inheritance and Polymorphism | Assignment 4 Due |
| 6    | Feb 25 | Chapter 10 Abstract Classes and Interfaces  
Chapter 11 Object-Oriented Design | Assignment 5 Due |
| 7    | Mar 4  | Chapter 12 Getting Started with GUI Programming  
Chapter 15 Creating User Interfaces | Assignment 6 Due |
| 8    | Mar 11 | **Spring Break** | |
| 9    | Mar 18 | **Mid Term Exam** | |
| 10   | Mar 25 | Chapter 14 Event Driven Programming  
Chapter 32 Java Database Programming | Assignment 7 Due |
| 11   | Apr 1  | Chapter 36 Remote Method Invocation | Assignment 8 Due |
| 12   | Apr 8  | Chapter 35 Java Server Pages | Assignment 9 Due |
| 13   | Apr 15 | Chapter 24 Multithreading  
Chapter 25 Networking | Assignment 10 Due |
| 14   | Apr 22 | Chapter 24 Advanced Swing Models  
Chapter 29 Menus, Toolbars, Dialogs | Assignment 11 Due |
| 15   | Apr 29 | Chapter 22 Containers, Layout Managers, and Borders | Assignment 12 Due |
| 16   | May 6  | Reading Day | Assignment 13 Due |
| 17   | May 13 | **Final Exam** | |