**Course Title: Core Java Programming II  
Course Number: CSE-40480  
Quarter: Fall 2016**

**Required Textbooks:**

***1) Introduction to Computing & Programming with Java***

 by Mark Guzdial and Barbara Ericson,

 Pearson/Prentice-Hall, ISBN:0131496980

 The software on the CD that comes with the textbook is also available online.

***You will find the most recent software version here:***

<http://coweb.cc.gatech.edu/mediaComp-teach#Java> (under **Java Book Resources**)

***2) Java How To Program, Tenth Edition***

 by Paul Deitel and Harvey Deitel,

 Pearson/Prentice-Hall, ISBN: 978-0-13-380780-6

**Instructor**

Name: W. Duane Wesley

E-Mail Address:   [SifuDuane@att.net](mailto:farid.naisan@mah.se) **Note: Communication within the Blackboard Discussions forum and Message facility is preferred.**

Virtual Office Hours: Flexible, will respond to email within 48 hrs.

**Teaching Philosophy**

The instructor uses PowerPoint slides to present the fundamental concepts, the basis of which can be found in recommended reading textbooks. The presentation will be made available to the students in PDF-format. The instructor believes that the best way of **learning** is **by** **doing,** and that **doing should be fun!** Therefore, the course requires the completion and submission of five projects that actively employ the knowledge learned from the course material.

**Goals**

The goal of the course is to cover the core elements and features of the Java language with emphasis upon object-oriented programming techniques. Students will be able to create fully functional applications in Java.

**Objectives**

The objectives of this course are the development of competencies in object-oriented concepts, Java language syntax, and Java application development. The course covers topics such as encapsulation, inheritance, dynamic binding, classes, interfaces, abstract classes, exceptions, collections, graphics, and graphical user interface components.

**Methods of Instruction**

Instruction is offered exclusively online; attendance or other physical presence is neither planned nor required. The course consists of lectures and practical training in the form of mandatory programming assignments. There will also be quizzes, but no comprehensive final exam. The assignments are done individually, although students are encouraged to discuss and exchange ideas with others.

A forum on the Blackboard course shell will be set up for asking questions, answering other students' questions, discussing and exchanging ideas, experiences and tips. Students are expected to participate actively in the discussions on the Blackboard shell.

**Grading**

Grades will be based entirely on the mandatory assignments, exercise, and quizzes. There will be no exams, other than the quizzes. All exercises will be assigned and completed during class. It is very important to maintain a good programming style and a good code structure. Remember that a satisfactorily functioning program combined with a well structured, documented and organized code project, will receive a higher grade than a fully functional program but with poor code structure and file organization.

**Grading Rubric**

|  |  |
| --- | --- |
| **Assessment** | **Weighting** |
| Quiz 1 | 4% |
| Quiz 2 | 4% |
| Quiz 3 | 4% |
| Quiz 4 | 4% |
| Quiz 5 | 4% |
| Assignment 1 | 10% |
| Assignment 2 | 10% |
| Assignment 3 | 10% |
| Assignment 4 | 10% |
| Assignment 5 | 10% |
| Assignment 6 | 10% |
| Assignment 7 | 10% |
| Assignment 8 | 10% |

**Outline**

The contents of the course are separated into five training modules. Every module will be complemented by PowerPoint slides, and will map to sections within the textbook.

The table below presents an outline of the course activities and weekly planning.  Typically, two weeks are devoted to each module. The schema may be adjusted according to the specific needs of the class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Modules** | **Topics** | **Textbook** | **Mandatory Projects** |
| **Module 1:** | Introduction  Objects and Classes  Types and Casting  Relational Expressions  Primitives  Strings  Variables and Memory Maps  Class and Object Methods  The CLASSPATH  Defining Classes and Creating Objects  Sending Messages to Objects  FileChooser and Picture  Playing a Sound  Object References vs. Primitives  Backslashes and Slashes | Chapters  1 - 3 | Assignment 1 Due  09/25/2016 |
| **Module 2:** | Arrays  Fields and Dot Notation  The For-Each Loop  The While Loop  Hierarchical Decomposition  Scope  The For Loop  Nested Loops  Method Return Values  Method Overloading  Conditional Execution | Chapters  4 - 6 | Assignment 2 Due  10/02/2016  Assignment 3 Due  10/09/2016 |
| **Module 3:** | Viewing the Java API  Packages  Inheritance  AWT Graphics  Interfaces  Static Methods  Private Methods  Another Way to Draw  Creating New Classes | Chapters  7 - 10 | Assignment 4 Due  10/16/2016  Assignment 5 Due  10/23/2016 |
| **Module 4:** | Defining a Class  Default Initialization  Constructors  Constructor Overloading  Using the Debugger  Setters and Getters  Javadoc Comments  Software Reuse via Inheritance  Dynamic Binding (Polymorphism)  ArrayLists  Working with Strings  Working with Files  Exceptions  The Random Class | Chapters   11 - 12 | Assignment 6 Due  10/30/2016 |
| **Module 5:** | Introduction to the Swing GUI Framework | Slides Only | Assignment 7 Due  11/13/2016  Assignments 8 Due  11/20/2016  (Yes, this is past the end of the course, but you can use this time if you need it.) |

**Reference Materials**

**Required**

See Textbook, above

**Links on the Web:**

<http://www.java.sun.com/>

**http://java.sun.com/docs/books/tutorial/uiswing/**

**http://www.javabeginner.com/java-swing/java-swing-tutorial**

**http://www.swingwiki.org/**

**Additional Learning Tools**

**JDK** (Java Development Kit) JDK 1.8.x (aka JDK 8)

**IDE:** DrJava or BlueJ and/or Eclipse or NetBeans