CompSci 262 Object Oriented Design & Programming II **Spring-17**

Instructor: George Georgiev

Office: HS, Room 217

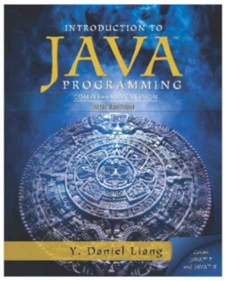
Phone: 424 - 11 80 **E-mail:** georgiev@uwosh.edu

Office Hours: M Tu W F: 9:10 – 10:10

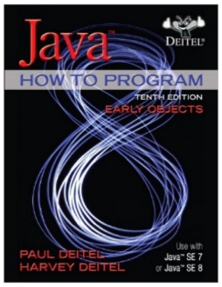
Lectures:	MTuW:	10:20 am - 11:20 am, HS 208
Labs:	F:	10:20 am – 11:20 am, HS 101 C

Recommended Text:

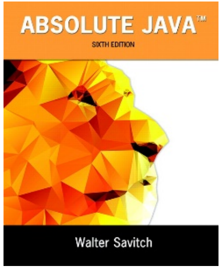
Introduction to Java Programming – Compressive version, 10th ed, Pearson, 2016.



Java – How to Program, 10th ed, Deitel & Associates, Inc., 2016.



Absolute Java, Savitch, Pearson, 6th ed., 2016.



All the materials and other information for the course will be posted on D2L.

Class Details

Status		Career	Undergraduate
Class Number	70018	Dates	1/30/2017 - 5/12/2017
Session	Fourteen Week	Grading	Student Option
Units	4 units	Location	Oshkosh
Instruction Mode	In Person	Campus	Main
Class Components	Lecture Required		

Current Catalog Description -
A second course in problem solving, software design, and computer programming using an object-oriented language. Problem solving/software design topics include: abstract data types, universal modeling language (UML), simple recursion, unit testing, event-handling, simple concurrency. Data structures and algorithms include: binary search, simple sorting algorithms, use of collection classes and their iteration protocols, sequential file processing. Additional topics include: inheritance, polymorphism, graphical user interfaces, simple use of threads. Prerequisites: Mathematics 108 or equivalent with a grade of C or better, or qualifying for a higher level mathematics course via the Mathematics Placement exam, and Computer Science 221 or equivalent with a grade of C or better. (Fall, Spring)

Prerequisite: Prerequisite: Mathematics 108 or equivalent with a grade of C or better, or qualifying for a higher level mathematics course via the Mathematics Placement exam, and Computer Science 221 or equivalent with a grade of C or better.

Course Outcomes
Topic areas and corresponding Learning Outcomes:

1. Debugging Java programs with BlueJ – You will be expected to:
    a. analyze a program on its correctness
    b. identify software bugs with the debugger in BlueJ
2. Objects and Classes – You will be expected to:
    a. specify a class with the UML graphical notation
    b. use the UML graphical notation to describe classes
    c. distinguish between object reference and primitive data type variables
    d. apply classes in the Java API (Application Programming Interface)
    e. differentiate between instance and static variables
    f. develop methods in classes
    g. store and process objects in arrays
    h. apply class abstraction to develop software
3. Inheritance and Polymorphism – You will be expected to:
    a. develop a subclass from a superclass through inheritance

   b. apply the polymorphism concept to handle different data types using a uniform interface

4. Abstract Classes and Interfaces – You will be expected to:
   a. identify the similarities and differences between an abstract class and an interface
   b. model weak inheritance relationships with interfaces
   c. specify a natural order using the Comparable interface
   d. to wrap primitive data values into objects
   e. create a generic sort method
   f. simplify programming using JDK 1.5 automatic conversion between primitive types and wrapper class types
5. Exceptions and Assertions – You will be expected to:
   a. distinguish exception types: Error versus Exception in Java
   b. throw an exception in a method
   c. write an exception handler using a try-catch-finally block
   d. explain the propagation of an exception
   e. apply assertions to help ensure program correctness
6. Text I/O – You will be expected to:
   a. read and write characters using the InputStreamReader, FileReader, BufferedReader, OutputStreamWriter, FileWriter, PrintWriter, BufferedWriter classes
   b. be able to apply the appropriate class in text I/O operations based on the requirements and performance needs
   c. distinguish between text I/O and binary I/O
7. Object-Oriented Design – You will be expected to:
   a. become familiar with the software development process
   b. model a system with the appropriate relationships: association, aggregation, composition, dependency, strong inheritance, and weak inheritance
   c. declare classes to represent the relationships among them
   d. design systems by identifying the classes and discovering the relationships among these classes
8. Unit testing with JUnit – You will be expected to:
   a. Create test classes, test methods, and run tests with JUnit
   b. Create and use test fixtures in JUnit
   c. Interpret test results with JUnit
   d. Correlate the test fixtures with assertions
   e. Verify that a software unit performs as specified
9. GUI and Graphics – You will be expected to:
   a. Describe the Java GUI hierarchy
   b. Create user interfaces using frames, panels, and simple GUI components
   c. Apply layout managers
   d. Use JPanel as subcontainers
   e. Draw figures using the methods in the Graphics class
   f. Override the paintComponent method to draw figures on a GUI component
   g. Introduction to Threads – creation, simple usage
10. Event Driven Programming – You will be expected to:
    a. declare listener classes and write event handlers to handle events
    b. apply the Observer Pattern to decoupled programs
    c. register listener objects in the source object

    d. create inner classes and anonymous inner classes
    e. write programs to handle ActionEvent, MouseEvent, KeyEvent, and Timer event

11. Recursion – You will be expected to:
    a. solve problems with recursion
    b. write program using recursion
    c. explain the difference between iteration and recursion
12. Generic Types – You will be expected to:
    a. improve reliability and readability of Java programs by using generic types
13. Java Collections Framework – You will be expected to:
    a. describe the Java Collections Framework hierarchy
    b. utilize the common methods in the Collection interface for operating sets and lists
    c. use the Iterator interface to traverse a collection
    d. examine the Set interface and be capable of deciding when to use HashSet, LinkedHashSet, or TreeSet to store elements
    e. compare elements suing the Comparator interface
    f. examine the List interface, and be capable of deciding how and when to use ArrayList or LinkedList to store elements
    g. examine the Collection and Map and be capable of deciding how and when to use HashMap, LinkedHashMap, and TreeMap to store values associated with keys.

Course Requirements:

There will be three exams, unannounced quizzes, programming assignments, and laboratory works. The material for all exams will come from either a material covered in class, lab work, and/or programming assignments.

Complete all required work on time. In the event that an exam must be missed, or required work can't be completed on time, due to illness or other serious and unavoidable circumstance, notify the professor in advance by phone or e-mail.

The programming assignments are due by 11:00 on the due date (electronic copy – drop box is due by 11:00 am, and a paper (hard) copy of the assignment is due at the beginning of class). Programs will be accepted up to three days late subject to the following penalties:

Turned in	Penalty
After 11:00 am on the due date	10%
1 day late	25%
2 days late	50%
3 days late	75%

Saturdays, Sundays, and holidays count when computing penalties.

If you work with partners, you will submit one electronic copy and one paper copy of the assignment with names on it. Partners will earn equal scores on the assignment. You may work alone on some assignments and with a partners on others. You may change partners during the semester.

You are encouraged to discuss assigned problems with other people but you must individually (or in group) design and write own solutions/code for all assignments. Submitting modified versions of other people's (group) work as your own is considered cheating.

There will be no make up for unannounced quizzes.

There will be one make up for the exams, which will cover all topics. It will be at the end of the semester.

Make up exam will be given if you call before the exam, make arrangements, have a medical certificate signed by the physician, and have a note from the Dean of Students Office.

The three exams will be announced at least a week before taking place.

Laboratory assignments will be in the teaching lab. The materials will be placed on D2L. You are encouraged to discuss the lab assignment with other students before and during the lab hours, but each student must demonstrate her or his own solution. If you do not finish a lab assignment during lab session, you have to demonstrate your solution to the instructor during the instructor's office hours before next lab.

Evaluation:

Three Exams:	~60%	(20% each)
Programming assignments:	~25%	(equal points for each assignment)
Unannounced quizzes:	~5%	(equal points for each quiz)
Laboratory Assignments	~10%	(equal points for each lab)

Grading:

Score	**Grade**
>= 92	A
90-92	A-
88-90	B+
82-88	B
80-82	B-
78-80	C+
72-78	C
70-72	C-
68-70	D+
62-68	D
60-62	D-
< 60	F

Feedback:

Your comments and questions about all aspects of the course (content, grading, teaching methods, pace, textbook, etc) are welcome. You can use e-mail or talk to me during office hours.