

**THE UNIVERSITY OF SOUTH CAROLINA LANCASTER**  
**Algorithmic Design II**  
**CSCE 146, Spring 2012**

**Contact Information:**

Professor: Dr. Noni Bohonak, HH 110

Phones: 803.313.7136 (office) 803.289-8697 (cell)

FAX: 803.313.7106

Office Hours: 9:00-10:00AM on M & W; 9:30-10:45AM on T & Th in HH110 with other times by appointment in the afternoon and evening or Fridays.

E-mail: [NBohonak@mailbox.sc.edu](mailto:NBohonak@mailbox.sc.edu)

Class Time: Jan 9 – Apr 23, 2012, 1:00-2:15pm, Monday and Wednesday;  
Monday Lab 2:30-4:30pm

Final Exam Time: Final Exam Monday, April 30<sup>th</sup>, 1-3:30pm

Location: HH222 on MW

**Course Information:**

Description: USC Online Bulletin (8/4/2011)

Continuation of CSCE 145. Rigorous development of algorithms and computer programs; elementary data structures.

**Credits:** 4-Credits

**Prerequisites:** Grade of C or better in CSCE 145 and grade of C or better in MATH 141

Note: Three lectures and two laboratory hours per week. Open to all majors.

**Texts:** Y. Daniel Liang, (2011). Introduction to Java Programming (8<sup>th</sup> Ed.), Upper Saddle River, NJ: Prentice-Hall.

**Learning Outcomes:**

1. Students should be able to describe, demonstrate, and apply the steps necessary to develop a working Object-Oriented program using Java in order to solve a problem.
2. Students should be able to apply design techniques and abstraction to reduce complex problems by dividing them into components that can be implemented in Java using methods and classes.
3. Students should be able to write clear, consistent, and unambiguous program documentation, both internally and externally.

4. Students should be able to write and debug programs to implement applications using:
  - objects and classes
  - strings and text I/O
  - inheritance and polymorphism
  - abstract classes and interfaces
  - simple GUI basics
  - event driven programming
  - user interfaces
  - applets and multimedia
  - exception handling
  - binary I/O
  - recursion
  - generics
  - Java collections framework
  - algorithm efficiency
  - lists, stacks, and queues
  - trees, iterators, heaps, and priority queues
  - sorting and searching
  - graph applications

### **General Education Goals:**

The General Education goals for the University of South Carolina at Lancaster (USCL) include a set of fundamental skills (reading, writing, reasoning, and oral communication), and the use of knowledge and thinking as a means to prepare students to assume a role in our present and future society. USCL divides these General Education goals into four areas: communication skills, critical thinking, cultural literacy, and student development. The purpose of this course is to help students meet these goals and the goals of the Computer Science Department by:

1. Developing structured, modular algorithms,
2. Implementing correct programs in an object-oriented language,

3. Using and implementing as classes data structures, such as sets, bags, sequence, stacks, queues, and binary trees,
4. Analyzing the time and space complexity of simple algorithms, and
5. Implementing moderately complex programs written in an object-oriented language.

## Grading:

A grade for the course will be determined by a calculation based on the following:

Computer Laboratory Assignments and Exercises	60%
Exams (Exam I, Exam II, Exam III, Final Exam)	40%

with the final grade based on the following scale:

90 – 100	A	70 – 75	C
86 – 89	B+	66 – 69	D+
80 – 85	B	60 – 65	D
76 – 79	C+	below 60	F

## Important Course Format Information

It is important that you check your USC email accounts several times during the week as all communications will be done through these accounts. Assignments will have due dates. You should submit these assignments on time during class or as instructed using Blackboard. Additional information will be posted using Blackboard.

## Assignments

Assignments will be made during class and/or posted on Blackboard. It is the responsibility of each student to check Blackboard at least once each week. Any missed work due to absences should be made up on time. Information will be posted on Blackboard at the end of class.

## Tentative Schedule

Week 1	Review of Objects and Classes, Strings and Text I/O
Week 2	Review continues

Week 3	Inheritance and Polymorphism
Week 4	Abstract Classes and Interfaces
Week 5	Object-Oriented Design and Patterns
Week 6	GUI Basics/ Graphics
Week 7	Event-Driven Programming/ Creating User Interfaces/Applets and Multimedia
Week 8	Event Handling and Binary I/O
Week 9	Recursion and Generics
Week 10	Java Collections Framework
Week 11	Algorithm Efficiency
Week 12	Lists, Stacks, and Queues
Week 13	Priority Queues
Week 14	Analysis of Algorithms
Week 15	Heaps and Sorts
Week 16	Graphs