GROSSMONT COLLEGE

Official Course Outline

COMPUTER SCIENCE INFORMATION SYSTEM 294 – INTERMEDIATE JAVA PROGRAMMING AND FUNDAMENTAL DATA STRUCTURES

1. Course Number Course Title Semester Units Semester Hours

CSIS 294 Intermediate Java Programming 4 3 hours lecture: 48-54 hours

and Fundamental Data Structures 3 hours lab: 48-54 hours

96-108 outside-of-class hours

for lecture

192-216 total hours

2. Prerequisites

A “C” grade or higher or “Pass” in CSIS 293 or equivalent.

Corequisite

None.

Recommended Preparation

None

3. Catalog Description

This course is the continuation of CSIS 293. Students will concentrate more on the Java techniques available for the development of large, complex systems. Students will utilize the concepts of Abstract Data Type to analyze real world requirements and design class structures to meet those requirements. In particular, students will apply these skills to the understanding and use of Data Structures. The course will progress from arrays, to linear lists, to stacks, queues, deques, and trees. Big-O~~h~~ notation will be introduced and used for the analysis and comparison of algorithms to perform sorting and searching of the structures. Students will become familiar with design techniques and tools (such as UML) necessary to develop larger programs. Instruction will also focus on object oriented programming and its principles of polymorphism, encapsulation, inheritance, collection classes and iteration protocols**.** Recursion and recursive data searching techniques will also be utilized in the creation of efficient, optimized algorithms.

4. Course Objectives

The student will:

1. Design and development programs using techniques (inheritance, polymorphism) necessary for development of large and complex systems that are extensible and/or reusable.
2. Utilize both custom and built-in data structures to solve computing problems.
3. Design and develop programs that implement and use linked lists**,** stacks, and queues.
4. Design and develop programs that implement and use trees both for data storage and data searching.
5. Utilize built-in data structures such as arrays and hash tables to optimize data storing and searching.
6. Utilize Generics and the creation of Generic types to build dynamic and flexible data structures.
7. Understand and be able to analyze requirements for particular searching and/or sorting algorithms.
8. Analyze the algorithmic performance of various algorithms.
9. Implement iterative or recursive solutions to various algorithms.
10. Analyze and choose the appropriate data structure for modeling a given problem.

5. Instructional Facilities

Standard computer lab with one internet-connected workstation per student with appropriate software installed

COMPUTER SCIENCE INFORMATION SYSTEM 294 – INTERMEDIATE JAVA PROGRAMMING AND FUNDAMENTAL DATA STRUCTURES Page 2

6. Special Materials Required of Student

Flash/USB drive or cloud storage for backup of in-class work.

7. Course Content

1. Use of inheritance and polymorphism in large system development.
2. Abstract data types.
3. Linked lists.
4. Stacks.
5. Queues and deques.
6. Trees.
7. Hash tables.
8. Recursive programming.
9. Searching and sorting algorithms.
10. Big-O notation and algorithm analysis.
11. Testing and debugging of complex algorithms.
12. Exception handling to include definition of own exception classes.
13. Generics and collections.
14. Design methodologies and tools (UML).

8. Method of Instruction

1. Lecture
2. Demonstration
3. Student exercises
4. Reading assignments

9. Methods of Evaluating Student Performance

a. Hands-on exercises.

b. Projects and lab activities.

c. Objective examinations and quizzes including a final examination.

10. Outside Class Assignments

a. Textbook reading assignments.

b. Prepare programming projects such as scientific, business, and action game programs in the Java programming language.

c. Algorithm and problem-solving exercises.

11. Texts

a. Required Text(s):

Goodrich, Michael T., Roberto Tamassia, and Michael H. Goldwasser. *Data Structures and Algorithms in Java.* 6th edition. Wiley Publishing, Indianapolis, IN. 2014

b. Supplementary texts and workbooks:

1) Deitel, Harvey and Paul Deitel. *Java: How to Program, Early Objects*. 11th edition. Boston, MA: Pearson, 2017.

2) Eckel, Bruce. *Thinking in Java*. 4th edition**.** Pearson/Prentice-Hall, New Jersey, 2006.

COMPUTER SCIENCE INFORMATION SYSTEM 294 – INTERMEDIATE JAVA PROGRAMMING AND FUNDAMENTAL DATA STRUCTURES Page 3

Addendum: Student Learning Outcomes

Upon completion of this course, our students will be able to do the following:

* 1. Given a complex problem specification, design and develop a Java program solution that successfully meets program requirements and demonstrates the appropriate and optimal application of data structures and object oriented techniques.
  2. Design and develop Java solutions that implement recursive algorithms including performing a binary search of sorted data and solving elementary factorial problems.
  3. Design and develop a Java program that uses generics to create custom, type-agnostic data structures modeled after linked lists, queues, stacks, and trees.

Date approved by the Governing Board: May 15, 2018