CSC 251  
ADVANCED JAVA PROGRAMMING

COURSE DESCRIPTION:

Prerequisites: CSC 151  
Corequisites: None

This course is a continuation of CSC 151 using the JAVA programming language with object-oriented programming principles. Emphasis is placed on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger. Upon completion, students should be able to design, code, test, debug, and implement objects using the appropriate environment. Course Hours Per Week: Class, 2. Lab, 3. Semester Hours Credit, 3.

LEARNING OUTCOMES:

Upon completion of this course, the student will be able to:

a. Apply object-oriented analysis and design to demonstrate key concepts and language structure of java.
b. Extend classes and apply the concept of inheritance  
c. Develop graphical user interfaces, taking advantage of layout managers supported by java technology.
d. Create threads that promote performing activities in parallel  
e. Design and develop a java application or applet utilizing multimedia techniques  
f. Design and develop a java application or applet that retrieves and updates information in a database.

OUTLINE OF INSTRUCTION:

I. Graphical User Interface
   A. Event-Driven Programming and Event Handling Model  
   B. Window Components  
   C. Mouse and keyboard event handling  
   D. Adapter classes  
   E. Layout managers

II. Graphics and Java 2D API
   A. Graphics contexts and graphics objects  
   B. Manipulate colors and fonts  
   C. Use classes Graphics and Graphics2D to draw lines, rectangles, rectangles with rounded corners, ovals, ellipse, polygons, arcs and general paths.  
   D. Specify Pain and Stroke characteristics of shapes.
III. Exception Handling
   A. Use try, throw, and catch to detect, indicate and handle exceptions
   B. Use the finally block to release resources
   C. Declare new Exception classes
   D. Uses stacks traces in debugging.

IV. Files and Streams
   A. Create, read, and update files using Java classes
   B. Use classes Scanner and Formatter to process text files
   C. Use the FileInputStream, FileOutputStream, ObjectInputStream, and ObjectOutputStream classes
   D. Use a JFileChooser dialog.

V. Searching and Sorting
   A. Use linear and binary search to search arrays
   B. Sort arrays using the iterative selection and insertion sort algorithms
   C. Sort arrays using the recursive merge sort algorithm.

VI. Data Structures
   A. Form linked data structures using references, self-referential classes and recursion.
   B. Create and manipulate dynamic data structures, such as linked lists, queues, stacks, and binary trees.
   C. Create reusable data structures with classes, inheritance and composition.

VII. Collections
   A. Use class Arrays for array manipulations
   B. Use the collections framework implementations
   C. Use the collections framework algorithms to manipulate collections.

VIII. Applets

IX. Multimedia
   A. How to get, display and scale images
   B. How to create animations from sequences of images
   C. How to create image maps
   D. How to get, play, loop and stop sounds, using an AudioClip
   E. How to play video using interface Player.

X. Multithreading
   A. Thread life cycle
   B. Thread priorities and scheduling
   C. Create and execute Runnables
   D. Thread synchronization
   E. Implement producer/consumer relationships
   F. Callable and Future interfaces.
XI. Networking
   A. Implement Java networking applications using sockets and datagrams
   B. Implement network-based applications
   C. Create a multithreaded server

XII. JDBC
   A. Relational database concepts
   B. Use Structured Query Language (SQL) to retrieve data from and manipulate data in a database.
   C. Use the JDBC API package (java.sql) to access databases.

REQUIRED TEXTBOOKS AND MATERIALS:

Text to be assigned by the instructor each semester.