Course Acronym:	CSCI
Course Number:	23
Descriptive Title:	Advanced Java Programming and the Android Operating System
Division:	Mathematical Sciences
Department:	Computer Science
Course Disciplines:	Computer Science
Catalog Description:	This course is an advanced Java programming language course that will include topics such as Graphical User Interface (GUI) development, multimedia, exception handling, network programming, multithreading, database manipulation, and Java servers and servlets. There will also be discussion about the Android operating system, including understanding the origins as well as how interaction is done with it. Students will complete a major Java programming project based on the topics taught in this course.
Prerequisite:	Computer Science 3 with a minimum grade of C or equivalent skill
Co-requisite:	
Recommended Preparation:	
Enrollment Limitation:	
Hours Lecture (per week):	3
Hours Laboratory (per week):	3
Outside Study Hours:	6
Total Course Hours:	108
Course Units:	4
Grading Method:	Letter Grade only
Credit Status:	Credit, degree applicable
Transfer CSU:	Yes
Effective Date:	Proposed
Transfer UC:	No
Effective Date:	
General Education: ECC	Area 4B - Language and Rationality: Communication and Analytical Thinking
Term:	
Other:	
CSU GE:	

Term:	
Other:	
IGETC:	
Term:	
Other:	
Student Learning Outcomes:	**SLO #1 SOFTWARE DEVELOPMENT** Design, code, compile, test and document programming solutions to problems requiring the development of Java classes (by inheritance, by composition; templates), requiring multimedia, graphics, multithreading, client/server programming, and network programming. **SLO #2 TRACING EXECUTION IN JAVA** When given a code segment involving use of a class, students will trace the construction and destruction of multithreaded programs, multimedia, and graphics. They will also verify whether memory leaks have occurred, as well as trace the connections of networks and databases. **SLO #3 ERROR CHECKING** When given Java code with errors, students will identify what those errors are and modify the Java code to eliminate those errors. **SLO #4 EXPLANATION OF CONCEPTS** Students will explain and differentiate among the concept of Java multimedia, graphics, multithreading, client/server programming, and network programming.
Course Objectives:	1. Develop Graphical User Interfaces to create Java applications using lightweight and platform-independent Java Swing components 2. Build multithreaded Java applications that allow several processes occurring simultaneously to be threaded into a single Java application 3. Build Java applications to create databases, database connectivity, and manipulate such databases 4. Write Java programs to handle program and application exceptions to improve robustness and reliability 5. Write web and intranet enabled Java applications running on Java servers and servlets, that serve the clients interacting with the server 6. Write Java applications that include multimedia, graphics, and animation software. 7. Use Remote Method Invocation (RMI) to implement three-tier client/server distributed applications 8. Create Java applications using enterprise classes, datagrams, and programming sockets via network protocols such as TCP/IP and UDP to communicate with Local Area Networks (LANs) 9. Explain how the Android Operating System was developed, including how Java relates to it.
Major Topics:	**I. Review of Java (3 hours, lecture)** A. Classes

B. Inheritance/Polymorphism

II. Graphics and Multimedia (9 hours, lecture)

A. Graphical User Interfaces (GUIs)
    1. Swing Interface Development
B. Audio
    1. Output of Sound Files
C. 2D Graphics
    1. Drawables
    2. Property Animation

III. Exception Handling (3 hours, lecture)

A. Overview of exception handling
B. Exception handling in Java

IV. Networking (6 hours, lecture)

A. Overview of Networking
    1. Protocols (TCP/IP, UDP, etc.)
    2. Topology
B. Network programming in Java

V. Database Connectivity (6 hours, lecture)

A. Overview of Databases
    1. Structured Query Language Lite (SQLite)
    2. Object-Oriented Database (DB)
    3. Java Connectivity to DB
B. Persistence
C. Cloud Storage
D. Application Security

VI. Multithreading (6 hours, lecture)

A. Overview of multithreading
B. Multithreaded programming in Java

VII. Client/Server Programming (6 hours, lecture)

A. Servlets
B. Java Server Pages
C. Remote Method Invocation (RMI)

VIII. Android Operating System (3 hours, lecture)

A. Overview of Android OS
B. Uses of Android OS

IX. Android Programming in Java (6 hours, lecture)

A. Overview
B. Programming in Android vs. programming in Java
C. Developer Tools
    1. Android Studio
    2. Gradle
    3. Emulators
D. Understanding the Model-View-Controller (MVC)
E. The Software Life Cycle in Android
F. Debugging Android programs

X. Group/Individual Project Presentations (6 hours, lecture)

XI. Review of Java (3 hours, lab)

A. Classes
B. Inheritance/Polymorphism

XII. Graphics and Multimedia (9 hours, lab)

A. Graphical User Interfaces (GUIs)
    1. Swing Interface Development
B. Audio
    1. Output of Sound Files
C. 2D Graphics
    1. Drawables
    2. Property Animation

XIII. Exception Handling (3 hours, lab)

A. Overview of exception handling
B. Exception handling in Java

XIV. Networking (6 hours, lab)

A. Overview of Networking
    1. Protocols (TCP/IP, UDP, etc.)
    2. Topology
B. Network programming in Java

XV. Database Connectivity (6 hours, lab)

A. Overview of Databases
    1. Structured Query Language Lite (SQLite)
    2. Object-Oriented Database (DB)
    3. Java Connectivity to DB
B. Persistence
C. Cloud Storage
D. Application Security

XVI. Multithreading (6 hours, lab)

A. Overview of multithreading

	B. Multithreaded programming in Java **XVII. Client/Server Programming (6 hours, lab)** A. Servlets B. Java Server Pages C. Remote Method Invocation **XVIII. Android Operating System (3 hours, lab)** A. Overview of Android OS B. Uses of Android OS **XIX. Android Programming in Java (6 hours, lab)** A. Overview B. Programming in Android vs. programming in Java C. Developer Tools 1. Android Studio 2. Gradle 3. Emulators D. Understanding the Model-View-Controller (MVC) E. The Software Life Cycle in Android F. Debugging Android programs **XX. Demonstration of Group/Individual Project Applications (6 hours, lab)**
Total Lecture Hours:	54
Total Laboratory Hours:	54
Total Hours:	108
Primary Method of Evaluation:	2) Problem solving demonstrations (computational or non-computational)
Typical Assignment Using Primary Method of Evaluation:	Write an activity that presents the user with a graphical interest calculator interface and calculates continuous compound interest for the given deposit, rate and time.
Critical Thinking Assignment 1:	This assignment has three parts: download a contact file, add data to Phone Contacts using code (not by Contact software), show the people's positions with their names on Google Map(you can only query the Contacts to get the data in this step). The Project should be named "MapOfContacts". Detail: a. Fetch the contact file "contact.txt" from the url: http://www.elcamino.edu/faculty/eambrosio/docs/cs45/contacts.txt The data will look like

	Dan dan@columbia.edu 40010787 116257324 John john@gmail.com 23079732 79145508 Daniel daniel@gmail.com 37985339 23716735 Johnny johnny@gmail.com 40774042 -73959961 Makiyo makiyo@gmail.com 36155618 139746094 Add first column to Name in the Contacts. Add second column to Email in the Contacts. Add third column to Mobile Number in the Contacts. (the number will be used as latitude in the Map) Add fourth column to Home Number in the Contacts. (the number will be used as longitude in the Map) b. You have to set up API keys in order to use Google Map API. You may use the following tutorial to set up your API keys: http://remwebdevelopment.com/dev/a35/Android-How-To-Set-Up-an-API-Key-forGoogle-Maps. c. Just showing markers on GoogleMap is not enough to show which person in what place, so I recommend you to use AlertDialog to show the name of the person when you tap it. d. This assignment has no restriction on what features you must use and also no restriction on what user interface you design. However, you must implement those three functions in code and be able to demonstrate that the results are correct.
Critical Thinking Assignment 2:	Write a menu driven application with one method that allows the user to input an Integer and prints out all factors of the argument and another method that allows the user to input an Integer and determines if the number is a prime number.
Other Evaluation Methods:	Completion, Homework Problems, Matching Items, Multiple Choice, Other Exams, Quizzes, True/False
Instructional Methods:	Demonstration, Discussion, Guest Speakers, Lab, Lecture, Role play/simulation
If other:	
Work Outside of Class:	Problem solving activity, Required reading, Study
If Other:	
Up-To-Date Representative Textbooks:	Y. Daniel Liang. Introduction to Java Programming and Data Structures, Comprehensive Edition. 12th ed. Pearson, 2021.
Alternative Textbooks:	
Required Supplementary Readings:	
Other Required Materials:	

Requisite:	Prerequisite
Category:	sequential
Requisite course(s): List both prerequisites and corequisites in this box.	Computer Science-3
Requisite and Matching skill(s):Bold the requisite skill. List the corresponding course objective under each skill(s).	**Develop algorithms using the Java programming language that involve translation of mathematical and algebraic steps, selection statements, iteration statements, and manipulate primitive data types and objects.** CSCI 3 - Identify Java data types. CSCI 3 -Develop programming code using control and iteration statements. **Use Java single and multi-dimensional arrays, List type data structures, and built in algorithms to design programming solutions requiring the storage and manipulation of large amounts of data (with random access ability during execution)** CSCI 3 -Develop programming code using strings and arrays, both single and multidimensional. **Use Java File manipulation and Data formatting Objects to format and store application result**. CSCI 3-Construct and use objects from predefined classes. CSCI 3 -Construct classes to encapsulate data and methods. CSCI 3 -Develop programming code using Input/Output files. **Understand Graphical User Interfaces and Event driven program design**. CSCI 3 Write and use static (class) and instance methods. CSCI 3 -Demonstrate ability to use inheritance and polymorphism in program code. CSCI 3 -Declare and use graphical components for user interfaces; handle basic mouse and keyboard events. CSCI 3 -Create and use applets with threads in world wide web applications.
Requisite Skill:	or equivalent skill
Requisite Skill and Matching Skill(s): Bold the requisite skill(s). If applicable	**Develop algorithms using the Java programming language that involve translation of mathematical and algebraic steps, selection statements, iteration statements, and manipulate primitive data types and objects.** Identify Java data types. Develop programming code using control and iteration statements. **Use Java single and multi-dimensional arrays, List type data structures, and built in algorithms to design programming solutions requiring the storage and manipulation of**

	large amounts of data (with random access ability during execution) Develop programming code using strings and arrays, both single and multidimensional. **Use Java File manipulation and Data formatting Objects to format and store application result**. Construct and use objects from predefined classes. Construct classes to encapsulate data and methods. Develop programming code using Input/Output files. **Understand Graphical User Interfaces and Event driven program design**. Write and use static (class) and instance methods. Demonstrate ability to use inheritance and polymorphism in program code. Declare and use graphical components for user interfaces; handle basic mouse and keyboard events. Create and use applets with threads in world wide web applications.
Requisite course:	
Requisite and Matching skill(s):Bold the requisite skill. List the corresponding course objective under each skill(s).	
Requisite Skill:	
Requisite Skill and Matching skill(s): Bold the requisite skill. List the corresponding course objective under each skill(s). If applicable	
Enrollment Limitations and Category:	
Enrollment Limitations Impact:	
Course Created by:	Greg Scott
Date:	09/07/2015
Original Board Approval Date:	
Last Reviewed and/or Revised by:	Edwin Ambrosio
Date:	04/27/2022
Last Board Approval Date:	06/20/2022 effective FA 2023