Course Outline for CS 34

ADVANCED JAVA PROGRAMMING

Effective: Fall

I. CATALOG DESCRIPTION:
CS 34 — ADVANCED JAVA PROGRAMMING — 4.00 units

This is an advanced course in Java programming. Content includes advanced topics that build on the skills acquired in earlier courses. It will cover advanced topics in Java: design and implementation of graphical user interfaces, exception handling, multithreading, database connectivity (JDBC), Servlets, networking, data structures, utility packages, collections, and JavaBeans.

3.00 Units Lecture 1.00 Units Lab

Prerequisite
CS 31 - Java Programming
with a minimum grade of C

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	54.00
Lab Hours:	54.00
Total Hours:	108.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:
Before entering the course a student should be able to:

  A. CS31

IV. MEASURABLE OBJECTIVES:
Upon completion of this course, the student should be able to:

  A. GENERIC: These outcomes are being developed throughout the entire programming sequence. Upon completion of the course, to an advanced level, students should be able to: Programming Skills
    1. Explain and use the design process
    2. Define and use functions and storage classes
    3. Define and explain trends in programming standards
    4. Write, compile, test and debug programs
    5. Present the characteristics of object-oriented programming
    6. Define and use constructor functions
    7. Define and use function overloading
    8. Define and use inheritance mechanisms in OOP
    9. Define and use user interfaces
    10. Define and use file I/O
    11. Define and develop class modules
    12. Develop and use event-driven programs
    13. Embed one language in another
  B. Database Design
    1. Explain database design concepts and the role of database components
    2. Create and customize forms and reports
    3. Explain the use of databases and information in the business environment
    4. Develop database business applications
  C. Technical Documentation
    1. Write in a concise and precise form appropriate for technical documentation
    2. Explain and use the processes and techniques of technical documentation
  D. Testing and Debugging
    1. Select debugging and testing methodology, and develop comprehensive and systematic test plan
    2. Design testing programs to uncover hardware compatibility problems during the development phase of the project
    3. Develop testing procedures

4. Conduct tests in the most efficient way
5. Test programs, and document errors and solutions
E. User Interface Design
    1. Define the requirements for the user interface
    2. Develop and test prototypes
    3. Construct user interfaces for flexibility and adaptability
F. Problem Solving
    1. Recognize a wide range of problems, and assess their impact on the system
    2. Use a wide range of troubleshooting methods and tools to isolate problems
G. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: Write programs that involve advanced Graphical User Interfaces.
H. Write programs that use exception handling.
I. Write programs that use multithreading.
J. Write programs that use multimedia.
K. Write programs using files and streams with Java Database Connectivity (JDBC)
L. Write programs using servlets.
M. Write programs using Remote Method Invocation (RMI).
N. Write programs using the network.
O. Write programs using simple data structures.
P. Write programs using Java Utilities Packages and bit manipulation.
Q. Write programs using Collections and Java Beans.

V. CONTENT:
  A. Advanced Graphical User Interfaces
  B. Exception Handling
  C. Multithreading
  D. Multimedia
  E. Files and Streams with Java Database Connectivity (JDBC)
  F. Servlets
  G. Remote Method Invocation (RMI)
  H. Networking
  I. Data Structures
  J. Java Utilities Packages and Bit Manipulation
  K. Collections
  L. JavaBeans

VI. METHODS OF INSTRUCTION:
  A. **Lecture** -
  B. **Demonstration** -
  C. **Projects** - Optional: Programming project completed in teams
  D. **Lab** - Lab Programming Assignments
  E. **Discussion** -

VII. TYPICAL ASSIGNMENTS:
A. Develop a graphical user interface (GUI). 1. This will collect personal data from a user to enroll in an organization. 2. The user will fill out a form, indicating name, address, city, state, zip, telephone number, and email address. 3. After the user presses the NEXT button, the program will perform complete verification of each data item, testing it for data type and plausibility of the value. 4. If there has been an erroneous input, the user will be notified by an error window indicating the nature of the error and be allowed to return to the data input screen for correction by pressing a CORRECT button. 5. Once the input data has been verified, the program will use it to update a database. 6. The program will then display a screen indicating the success of the update and offer the user a chance to exit by pressing a DONE button. B. Write a program to create a database of customers: 1. The program is to be a GUI application 2. The customer record contains the following data a. Name (20 char) b. Address (50 char) c. Phone (20 char) d. Number of Employees e. Average salary of employees 3. The file of customers is to be a binary file where data is stored as a DLL 4. The application should be capable of creating, modifying, and deleting all customer records. C. Write a team programming project to create a WEB application that will act as a web browser and as a database which contains users and their web sites visited: 1. The browser application behaves just as normal browsers behave but that each user is added to the list of users and the web sites visited are tracked within the database. 2. The system administrator to the Web application can view the list of web sites and disable selected sites. 3. When a user starts the application and attempts to visit a restricted site they are denied access.

VIII. EVALUATION:
  A. **Methods**

  B. **Frequency**

    1. Frequency of evaluation
      a. Recommend 2 or 3 exams plus final examination
      b. Recommend programming assignment to cover each topic within course content. Contents can be combined.
    2. Types of Exam Questions
      a. Determine what is calculated by the given recursive method:

```
public static int f(int n)
{
   if (n == 0)
      return 0;
   else
      return n* f(n - 1);
}
```

    1. Write a program to concatenate two text files: that is, to produce a file that contains the first file followed by the second file.
    2. Design a class hierarchy for the class of geometric figures, starting with the base class Point and derived classe Circle from which is derived the class Ellipse. Include suitable data members, and member functions. Write the code and make a diagram showing the inheritance relationships.

IX. TYPICAL TEXTS:
  1. Mary Campione and Kathy Walrath *The Java Tutorial.* 2nd ed., Addison Wesley, 2001.
  2. van der Linden *Just Java* 2., Sun Microsystems Press, 1999.

3. Lewis and Berg *Multithreaded Programming with Java Technology.*, Sun Microsystems Press, 2000.
4. Geary *Graphics Java 2, Volume II Swing.*, Sun Microsystems Press, 1999.
5. Deitel and Deitel *Java How to Program.* 3rd ed., Prentice Hall, 1999.
6. Joe Wigglesworth and Paula Lumby *Java Programming Advanced Topics.*, Course Technology, 2000.

X. OTHER MATERIALS REQUIRED OF STUDENTS:
A. Current version of Borland J Builder
B. Current version of Microsoft Visual J++
C. Current version of Symantec Visual Café