



**COWLEY COLLEGE
& Area Vocational Technical School**

COURSE PROCEDURE FOR

**ADVANCED JAVA PROGRAMMING
CIS1870 3 Credit Hours**

Student Level:

This course is open to students on the college level in either freshman or sophomore year.

Catalog Description of the Course:

CIS1870 - ADVANCED JAVA PROGRAMMING (3 hrs)

An introductory course to give computer science majors an introduction to programming in Java. Advanced topics will be covered: Swing to create graphical user interfaces, the model-view-controller, graphics programming, JavaBeans, concepts of security, Java Database Connectivity, servlets, JavaServer pages, RMI, EJBs, JMS, and Jini.

Prerequisite:

CIS1868 Java Programming

Controlling Purpose:

This course is offered to teach advanced programming skills in Java to students. The topics learned in this course will allow the student to explore some of the various uses for Java.

Learner Outcomes:

Upon completion of this course, the student should be able to write more advanced Java applications. The student will learn Swing to create graphical user interfaces, the model-view-controller, graphics programming, javabeans, concepts of security, Java Database Connectivity, servlets, JavaServer pages, RMI, EJBs, JMS, and Jini. Some of these concepts will be taught but not necessarily programmed.

Units Outcomes and Criterion Based Evaluation Key for Core Content:

The following defines the minimum core content not including the final examination period. Instructors may add other content as time allows.

Evaluation Key:

- A = All major and minor goals have been achieved and the achievement level is considerably above the minimum required for doing more advanced work in the same field.
- B = All major goals have been achieved, but the student has failed to achieve some of the less important goals. However, the student has progressed to the point where the goals of work at the next level can be easily achieved.
- C = All major goals have been achieved, but many of the minor goals have not been achieved. In this grade range, the minimum level of proficiency represents a person who has achieved the major goals to the minimum amount of preparation necessary for taking more advanced work in the same field, but without any major handicap of inadequacy in his background.

- D = A few of the major goals have been achieved, but the student's achievement is so limited that he is not well prepared to work at a more advanced level in the same field.
- F = Failing, will be computed in GPA and hours attempted.
- N = No instruction or training in this area.

CHAPTER 2: ADVANCED SWING GRAPHICAL USER INTERFACE COMPONENTS						
Outcomes: Upon completion of this unit, the student will understand how to design Java GUI's using advanced swing components						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Be able to use Swing components to enhance application GUIs.
						Be able to use Swing text components to view styled documents.
						Understand the Command design pattern and its implementation in Swing.
						Be able to develop applications with multiple-document interfaces.
						Understand how to implement drag-and-drop support.
						Learn how to prepare internationalized applications.
						Understand how to use Swing to create accessible applications for people with disabilities.

CHAPTER 3: MODEL-VIEW-CONTROLLER						
Outcomes: Upon completion of this unit, the student will understand how to use model-view-controller architecture to represent data in an application						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand the model-view-controller (MVC) architecture for separating data, presentation and user input logic.
						Understand the Observer design pattern.
						Understand MVC's use in Java's Swing GUI components.
						Understand the default model implementations for Swing components.
						Understand the use of TableModels to represent tabular data for Jtables.
						Understand tree data structures and their use as TreeModels for Jtrees.

CHAPTER 4: GRAPHICS PROGRAMMING WITH JAVA 2D AND JAVA 3D

Outcomes: Upon completion of this unit, the student will understand how to use the Java 2D and 3D API to draw basic shapes

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Be able to use the Java 2D API to draw various shapes and general paths.
						Be able to specify Paint and Stroke characteristics of shapes displayed with Graphics2D.
						Be able to manipulate images using Java 2D image processing.
						Use the Java 3D API and Java 3D Utility classes to create three-dimensional graphics scenes.
						Manipulate the texture and lighting of three-dimensional objects with Java 3D.

CHAPTER 6: JAVABEANS COMPONENT MODEL

Outcomes: Upon completion of this unit, the student will understand how to write programs that implement JavaBeans

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand JavaBeans and how they facilitate component-oriented software construction.
						Understand how to use Forte for Java Community Edition to build JavaBeans-based applications.
						Be able to wrap class definitions as JAR files for use as JavaBeans and stand-alone applications.
						Be able to define JavaBean properties and events.

CHAPTER 7: SECURITY

Outcomes: Upon completion of this unit, the student will understand how to write programs that implement basic security and cryptography

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand the basic concepts of security.
						Understand public-key/private-key cryptography.
						Learn about popular security protocols, such as SSL.
						Understand digital signatures, digital certificates and certification authorities.
						Learn how Java provides solutions to security problems.
						Learn how to produce secure code with Java technology.

CHAPTER 8: JAVA DATABASE CONNECTIVITY (JDBC)

Outcomes: Upon completion of this unit, the student will understand how to write programs that access databases through the Java Database connectivity package

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand the relational-database model.
						Understand basic database queries using Structured Query Language (SQL).
						Use the classes and interfaces of package java.sql to manipulate databases.
						Use transaction processing to prevent database updates from modifying the database if an error occurs during a transaction.
						Introduce the JDBC 2.0 optional package javax.sql's capabilities for obtaining database connections, creating connection pools and treating result sets as JavaBeans.

CHAPTER 9: SERVLETS

Outcomes: Upon completion of this unit, the student will understand how to write programs that run on the server

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand how to execute servlets with the Apache Tomcat server.
						Be able to respond to HTTP requests from an HttpServlet.
						Be able to redirect requests to static and dynamic Web resources.
						Be able to maintain session information with cookies and HttpSession objects.
						Be able to access a database from a servlet.

CHAPTER 10: JAVASERVER PAGES (JSP)

Outcomes: Upon completion of this unit the student will understand how to write javaserver pages

A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Be able to create and deploy JavaServer Pages.
						Use JSP's implicit objects and Java to create dynamic Web pages.
						Specify global JSP information with directives.
						Use actions to manipulate JavaBeans in a JSP, to include resources dynamically and to forward requests to other JSPs.
						Create custom tag libraries that encapsulate complex functionality in new tags that can be reused by JSP programmers and Web-page designers.

CHAPTER 13: REMOTE METHOD INVOCATION						
Outcomes: Upon completion of this unit the student will understand how to write basic programs that implement distributed services						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand the distributed computing concepts.
						Understand the architecture of RMI.
						Be able to use activatable RMI objects to build resilient distributed systems.
						Understand how to use RMI callbacks.
						Be able to build RMI clients that download necessary classes dynamically.
						Be able to build activatable RMI objects.

CHAPTER 14: SESSION EJBS AND DISTRIBUTED TRANSACTIONS						
Outcomes: Upon completion of this unit, the student will understand how to write basic programs that process distributed transactions						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand EJBS as business-logic components.
						Understand the advantages and disadvantages of stateful and stateless session EJBS.
						Understand JNDI's role in enterprise Java applications.
						Understand distributed transactions.
						Understand the advantages and disadvantages of container-managed and bean-managed transaction demarcation.

CHAPTER 15: ENTITY EJBs						
Outcomes: Upon completion of this unit the student will understand the concept of entity EJs						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand how entity EJBs represent persistent data.
						Understand synchronization issues between EJBs and database data.
						Understand the life-cycle of an entity EJB.
						Understand the advantages and disadvantages of container-managed and bean-managed persistence.

CHAPTER 16: MESSAGING WITH JMS						
Outcomes: Upon completion of this unit the student will understand how to write programs that implement the Java Message Service API						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand message-oriented middleware.
						Understand the point-to-point messaging model.
						Understand the publish/subscribe messaging model.
						Understand the difference between the two messaging models and when it is appropriate to use each.
						Understand how to use the Java Message Service (JMS) API to build messaging applications in Java.
						Introduce message-driven EJBs.

CHAPTER 22: JINI						
Outcomes: Upon completion of this unit, the student will understand how to write programs that implement Jini Technology Architecture						
A	B	C	D	F	N	Specific Competencies
						Demonstrate the ability to:
						Understand the Jini Technology Architecture.
						Be able to identify the major components of a Jini solution.
						Be able to implement Jini services and register those services with Jini lookup services.
						Be able to write a Jini client.
						Be able to use Jini helper classes to simplify service implementations.

Projects Required:

Projects will vary according to the instructor.

Text Book:

Contact bookstore for current textbook information.

Materials/Equipment Required:

None

Attendance Policy:

Students will adhere to the attendance policy outlined by the instructor in the course syllabus.

Grading Policy:

Grading procedures will vary according to the instructor.

Maximum Class Size:

Based on classroom occupancy

Course Time Frame:

The U.S. Department of Education, Higher Learning Commission and the Kansas Board of Regents define credit hour and have specific regulations that the college must follow when developing, teaching and assessing the educational aspects of the college. A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work for approximately fifteen weeks for one semester hour of credit or an equivalent amount of work over a different amount of time. The number of semester hours of credit allowed for each distance education or blended hybrid courses shall be assigned by the college based on the amount of time needed to achieve the same course outcomes in a purely face-to-face format.

Refer to the following policies:

[402.00 Academic Code of Conduct](#)

[263.00 Student Appeal of Course Grades](#)

[403.00 Student Code of Conduct](#)

Disability Services Program:

Cowley College, in recognition of state and federal laws, will accommodate a student with a documented disability. If a student has a disability which may impact work in this class which requires accommodations, contact the Disability Services Coordinator.