Des Moines Area
Community College

COURSE COMPETENCY INFORMATION

Effective Date: Fall 2016

Acronym/Number: **CIS 175** **Historical Reference:**

Course Title: **Java II**

Credit Breakout:	**3.00**	**3.00**	**0**	**0**	**0**
	(Total Credits:	**Lecture** *1hr/crdt*	**Lab** *2hrs/crdt*	**Practicum** *3hrs/crdt*	**Work Experience)** *4hrs/crdt(unsupervised)*

PREREQUISITE/CO-REQUISITE

Pre-requisite: CIS-171

DESCRIPTION

This course is a continuation of Java. Additional concepts of object-oriented programming will be applied in a variety of programming exercises.

COURSE COMPETENCIES

1. Construct Handling Exceptions into Java Programs
    1. Differentiate among checked exceptions, unchecked exceptions, and errors
    2. Create a try-catch block and determine how exceptions alter normal program flow
    3. Describe the advantages of Exception handling
    4. Create and invoke a method that throws and catches an exception
    5. Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)
2. Create Advanced Graphical User Interfaces using Swing
    1. Implement simple graphical user interfaces
    2. Add buttons, text fields, and other components to a frame window
    3. Handle events that are generated by buttons
    4. Write programs that display simple drawings
    5. Use layout managers to arrange user-interface components in a container
    6. Demonstrate familiarity with common user-interface components, such as radio buttons, checkboxes, and menus
    7. Build programs that handle events generated by user-interface components
3. Identify methods of deploying applications
    1. Describe how the following items are deployed: Individual .class files, JAR files, OS wrapper around class files or JAR files, Applets, Java Web Start, WAR files

    2. Discuss server-based alternatives to deploying applications

4. Build a Database Application with JDBC
    1. Understand how relational databases store information
    2. Read and understand a UML diagram
    3. Describe the interfaces that make up the core of the JDBC API including the Driver, Connection, Statement, and ResultSet interfaces and their relationship to provider implementations
    4. Identify the components required to connect to a database using the DriverManager class including the JDBC URL
    5. Query a database with the structured Query language (SQL)
    6. Connect to a database with Java Database Connectivity (JDBC)
    7. Write database programs that insert, update, and query data in a relational database
    8. Submit queries and read results from the database including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections
    9. Explain the purpose and benefits of an object/relational mapping tool
5. Create a Java Server Page and Servlet
    1. Describe the role of JSP and servlets
    2. Describe the basic structure of servlets
    3. Create a servlet that generates plain text
    4. Create a servlet that generates HTML
    5. Describe the servlet life cycle
    6. Illustrate servlet debugging strategies
    7. Understand the need for JSP and evaluate the benefits
    8. Understand the JSP lifecycle
    9. Create and install basic JSP pages
6. Create a basic script
    1. Select a scripting engine for the appropriate circumstance
    2. Create and execute a basic script
    3. Discuss the advanced features scripting engines offer
7. Examine Java Application Security
    1. Distinguish the security concerns that surround application development
    2. Discuss the three mechanism to help ensure safety of Java code: language, access control and code signing
    3. Illustrate how digital signatures are utilized
    4. Explain code signing
    5. Explain how encryption works and why it is important
    6. Sanitize user input
8. Utilize version control
    1. Summarize how to setup the Git environment
    2. Demonstrate how to set up a new project and repository
    3. Explain how to work with code, including checking out, committing changes, destroying unwanted changes, viewing revisions and history
9. Discuss Agile and Development Approaches
    1. Describe the players in the agile environment

2. Explain the Agile development process and components, including user stories, test-driven development, SCRUM, sprints and backlog
3. Identify the advantages and disadvantages of various Agile development methods, including Kanban, Scrum and Extreme Programming
4. Identify other methodology of software development processes, including rapid application, prototyping and waterfall approaches
5. Understand the components that go into full stack development