

Course Justification

"The Java platform has attracted over 5 million software developers, worldwide use in every major industry segment, and a presence in a wide range of devices, computers, and networks of any programming technology. ...you can find Java technology in networks and devices that range from the Internet and scientific supercomputers to laptops and cell phones, from Wall Street market simulators to home game players and credit cards -- just about everywhere." (Java Technology: Brief History of Java Technology, <http://java.com/en/about/>)

In addition to widespread use in industry, the Java language has also found wide adoption in Computer Science education with many colleges requiring it as part of the Computer Science, Computer Information System and Information Technology degree programs.

COP 2800 – Java Programming

Final 3/11/2008

Course Description – This course is an intermediate programming course using the Java computer language. Students code, compile and execute programs using some advanced programming concepts and object oriented programming and design concepts and principles. Prerequisites: COP 1220 Laboratory fee. (3hr. lecture; 2hr lab)

Course Competencies

Competency 1: The student will demonstrate an understanding of the Java system architecture and its major components by:

- a. Distinguishing between the Java Runtime Environment (JRE) and the Java Development Kit (JDK)
- b. Identifying the Java Virtual Machine (JVM) and the Java compiler
- c. Describing the process of coding, compiling and running from the command line.
- d. Differentiating between *.java and *.class files
- e. Successfully installing the JDK and compiling a program from the command line that uses at least one optional Java package.
- f. Using the JDK standard packages and API documentation in developing their own programs.

Competency 2: The student will demonstrate an understanding of the professional software development process by:

- a. Designing and documenting solutions at the method level by writing pseudocode or developing flow charts for development before writing the code.
- b. Designing and documenting solutions at the project level by using an object-oriented design technology such as UML or CRC cards.
- c. Coding software solutions following professional coding style guidelines
- d. Incorporating adequate and meaningful comments into the source code of programming projects using both standard and *javadoc* style comments
- e. Testing and designing tests of software solutions.
- f. Debugging program code.

Competency 3: The student will demonstrate an understanding of fundamental programming constructs and concepts by:

- a. Using appropriate data types for programming assignments.
- b. Using Boolean, comparison, arithmetic and object (instanceof) operators in their programs
- c. Explaining the properties of a variable such as its name, value, scope, persistence, and size.
- d. Distinguishing between expressions and statements.
- e. Identifying and using the three control structures (sequence, selection and repetition).

Competency 4: The student will demonstrate an understanding of the following advanced programming techniques by:

- a. Parsing a string and using other string manipulation techniques.
- b. Using both arrays and the Java collections to process aggregate data.
- c. Using object composition (object references) to build more complex objects.
- d. Developing an event driven program.
- e. Writing a recursive algorithm for solving a problem and identifying its exit condition.

Competency 5: The student will demonstrate an understanding of the object oriented programming concepts of *Class* and *Object* by:

- a. Identifying and using *instance variables* and *instance methods*.
- b. Using, programming and identifying *constructors*
- c. Explaining the process of object instantiation
- d. Using, programming and identifying *accessor* and *mutator* methods
- e. Using, programming and identifying *class (static) variables* and *class (static) methods*.
- f. Using, programming and identifying *overloaded* methods and constructors.
- g. Creating programs using inner classes and describing their effects on generated class files.

Competency 6: The student will demonstrate an understanding of inheritance by:

- a. Explaining the benefits of inheritance.
- b. Creating a class which *extends* a parent class.
- c. Explaining the restrictions imposed when using inheritance.
- d. Overriding and overloading parent class functions within a child class.
- e. Distinguishing between inheritance of implementation (*extends*) and inheritance of design (*implements*)
- f. Creating a class which implements an *interface*
- g. Creating a class which extends an *abstract class*

Competency 7: The student will demonstrate an understanding of Object Oriented Design concepts by:

- a. Using visibility modifiers (public, private, protected) to implement appropriate *abstraction* and *encapsulation*.
- b. Explaining *coupling* and how to achieve loose coupling.
- c. Explaining *cohesion*, and how to achieve high cohesion.
- d. Writing a program which demonstrates *polymorphism*.

Competency 8: The student will demonstrate an understanding of Java input and output by:

- a. Describing I/O
- b. Creating programs that use console I/O
- c. Creating programs that GUI (dialog box) I/O
- d. Creating programs that use file I/O

Competency 9: The student will demonstrate an understanding of exception programming techniques by:

- a. Describing exceptions
- b. Encapsulating exceptions
- c. Throwing and catching exceptions

COP 2800 - Java Programming

Final 3/11/2008

Competency 10: The student will demonstrate an understanding of GUI (Graphical User Interfaces) and event driven programs by:

- a. Describing the Java GUI library structure (AWT and swing).
- b. Writing a GUI program using a single window and appropriate components.
- c. Describing the uses of layout managers.
- d. Creating programs that handle events.

Competency 11: The student will demonstrate an understanding of professional development by:

- a. Participating as a member of a project team.
- b. Finding and reviewing a professional source of information.