Catalog Data: Object Oriented Programming with Web-Based Applications (3) This course will continue with object-oriented programming and will introduce graphics, virtual machines, programming language concepts and web-based programming using Java.
Prerequisites: CMPSC 121, CMPSC 122.


At University Park, the primary resource is a series of lecture notes developed by Doug Hogan, available at http://www.personal.psu.edu/djh300/cmpsc221-general/.

Course Objectives: CMPSC 221 is the third in the introductory programming sequence and begins by transitioning students from C++ to Java. Then it proceeds with three major units of study:

- **Object-Oriented Programming**: This unit builds upon the object-based programming learned in CMPSC 122 and introduces the students more advanced aspects of the object-oriented paradigm, especially inheritance and polymorphism.

- **Graphics and Graphical User Interfaces**: This unit teaches students how to use Java’s built-in drawing tools to create graphics that scale to containers of any size. It also introduces students to creating graphical user interfaces and doing layouts to make content scale to windows. In the process, the event-driven programming paradigm is introduced and the emphasis is on creating applets that can be used on the web.

- **Databases**: In the final unit, students begin by learning high-level concepts of relational databases and study the basics of SQL, emphasizing retrieving data from database tables. Students then learn how to incorporate database code in Java via Java Database Connectivity (JDBC) and create GUI front ends for databases driven by SQL.

Throughout the course, there is an emphasis on producing reusable code and generalize-able code, especially writing preconditions and postconditions for all methods. In addition, creating large projects that tie together several of the programming skills is a vital component. In particular, the final project must involve creating GUIs with a database-driven back-end and a project must include designing and implementing an inheritance hierarchy to solve a problem.

Primary Course Outcomes: Upon completion of the course, students should be able to successfully complete:

- Correctly implement a class in the target language, including the appropriate use of public and private methods and data.
- Implement a subclass of the existing class in order to extend the functionality of the base class in order to solve the stated problem.
- Create programs which draw graphics dynamically to scale to the size of a window.
- Create GUI programs with event handling and with layouts that scale to the size of a window.
- Create GUI programs which incorporate exception handling.
- Create database tables and query them in SQL, including queries which go across more than one table.
- Create GUI programs in Java which act as the front-end to a database whose back-end is implemented in SQL.
Relationship to Undergraduate Program Outcomes:

CMPSC 221 is the third in a sequence of programming courses which collectively support the following program outcomes:

- Develop a modest (on the order of a thousand lines of code) software application, using appropriate data structures and algorithms.
- Analyze algorithms or computer code for correctness and efficiency.

Required Topics:

- Unit 0: Course Introduction and Transition to Java (3 classes)
  - The organization of Java and virtual machines
  - Similarities and differences between Java and C++
  - Input and output in Java
  - Primitive and reference types in Java, including arrays
  - Methods in Java
  - Random numbers
  - Review of basic control structures

- Unit 1: Object-Oriented Programming (8 classes)
  - Classes vs. objects
  - Access rights
  - Constructors, modifiers, accessors
  - this references
  - Garbage collection
  - static data and methods
  - final data
  - Composition of classes
  - External documentation of methods
  - Inheritance
  - protected vs. private
  - super
  - Inheritance hierarchies and "is-a" and "has-a" relationships
  - Polymorphism
  - Abstract methods and classes
  - Java's exception hierarchy
  - Exception handling

- Unit 2: Graphics and Graphical User Interfaces (9 classes)
  - Pixels and window organization
  - Primitive Java drawing tools: text, lines, rectangles, ovals, arcs
  - Colors and the RGB model
  - Applets
  - Embedding applets in HTML
  - GUI front-end elements: labels, text fields, buttons, text areas, checkboxes, radio buttons, drop down lists, lists
  - GUI layout: panels and layout managers, especially BorderLayout
  - Event Handling
  - Event handlers for semantic events: ActionListener, ItemListener
  - Event handlers for hardware events: KeyListener, MouseListener
  - Intermediate graphics: changing colors and fonts, polygons and polylines
  - Inner classes for event handling
  - Advanced GUI elements such as sliders and menus
  - Using the Java API to learn about GUI elements
• Unit 3: Databases (8 classes)
  • Overview of relational database organization, primary keys, and foreign keys
  • SQL overview
  • Creating tables in SQL
  • Inserting data in SQL
  • Retrieving data in SQL with one-table select statements, including group and group filters
  • Updating and deleting records in SQL
  • Creating foreign key relationships in SQL.
  • Querying across multiple tables with subqueries
  • JDBC

Class Format: Two lecture/labs per week. Each lecture/lab is 75 minutes.

Professional Component: CMPSC 221 is designed to aid in the professional development of engineers and scientist by developing skills in problem solving, critical thinking, algorithm design, and program implementation. Although Java and SQL will be used to demonstrate these skills, many of the concepts can be applied to general engineering problems and writing programs in other languages as well.

Evaluation: Evaluation will be by examinations and programming projects. Roughly 50% of the grading comes from proctored examinations (and, optionally, quizzes) and the other 50% comes from programming assignments. About half the programming grade comes from short lab assignments reinforcing the content of one or two classes and about half the programming grade comes from large projects that tie together all of the skills in one or more units or the course.

Author: Al Verbanec, Doug Hogan
Last Revised: November 14, 2012