

COURSE SYLLABUS

LAST REVIEW	Spring 2021
COURSE TITLE	Object Oriented Programming using Java
COURSE NUMBER	CIST-0256
DIVISION	Career and Technical Education
DEPARTMENT	CIST
CIP CODE	24.0101
CREDIT HOURS	3
CONTACT HOURS/WEEK	Class: 3 Lab: 0
PREREQUISITES	CIST-0180: Programming Algorithms
COREQUISITES	None

COURSE DESCRIPTION

This course is designed as a continuation of course CIST-0180. This course will include a brief review of the material covered in CIST-0180 and then proceed to address the more advanced features of the Java programming language. Programs and projects are designed to illustrate the use of classes, objects, Inheritance, Polymorphism and Composition. Java GUI elements will be examined in detail. The life cycle and use of applets (with graphics) will be reviewed along with screen management. Event driven processing, multithreading and exception handling will be demonstrated by class projects. The student must have successfully completed course CIST-0180, Java with a grade of "C" or better.

PROGRAM ALIGNMENT

This course is part of a program aligned through the Kansas Board of Regents and Technical Education Authority. For more information, please visit:

https://kansasregents.org/workforce_development/program-alignment

PROGRAM LEARNING OUTCOMES

1. Develop Java applications: Develop a complete Java procedural application.
- 2.

INSTITUTIONAL LEARNING OUTCOMES

- Communication
- Computation and Financial Literacy
- Critical Reasoning
- Technology and Information Literacy
- Community and Civic Responsibility
- Personal and Interpersonal Skills

TEXTBOOKS

<http://kckccbookstore.com/>

METHOD OF INSTRUCTION

A variety of instructional methods may be used depending on content area. These include but are not limited to lecture, multimedia, cooperative/collaborative learning, labs and demonstrations, projects and presentations, speeches, debates, panels, conferencing, performances, and learning experiences outside the classroom. Methodology will be selected to best meet student needs.

COURSE OUTLINE

- I. Java Programming Language
 - A. Object-oriented programming in Java
 - B. Java program structure
 - C. Elementary of java programming
- II. Defining class
 - A. What is a class?
 - B. Constructors
 - C. Defining and using a class
 - D. Method overloading
 - E. Multiple constructors
 - F. Using objects
 - G. Controlling access to class members
 - H. Inner classes
- III. Extending Classes and Inheritance
 - A. Using existing classes
 - B. Choosing base class access attributes
 - C. Polymorphism
 - D. Multiple levels of inheritance
 - E. The object class
 - F. Casting objects
 - G. Designing classes
 - H. Abstract classes
 - I. Using the final modifier
 - J. Interfaces
- IV. Exceptions
 - A. The idea behind exceptions
 - B. Types of exceptions
 - C. Specifying the exceptions a method can throw
 - D. Handling exceptions
 - E. Dealing with exceptions
- V. Graphic User Interface (GUI)

- A. Frames
- B. Containers
- C. Compartments
- D. Graphics
- E. Layout Management
- F. Panels
- G. Event Handling
- VI. Multithreading
 - A. Thread class
 - B. Thread's life cycle

COURSE LEARNING OUTCOMES AND COMPETENCIES

Upon completion of the course, the student will:

- A. Show knowledge of object-oriented programming in Java, define Java program structure, define the use of packages, controlling access to class members and inner classes.
 1. Show knowledge of object-oriented programming in Java.
 2. Define Java program structure.
 3. Define the use of packages.
 4. Define controlling access to class members and inner classes.
- B. Explain extending classes and inheritance, using existing classes and choosing base class access attributes, and explain polymorphism.
 5. Explain extending classes and inheritance.
 6. Explain using existing classes and choosing base class access attributes.
 7. Explain polymorphism.
- C. Explain multiple levels of inheritance, use super classes that have constructors or arguments, and define designing and abstract classes.
 8. Explain multiple levels of inheritance.
 9. Use super classes that have constructors.
 10. Use super classes that require arguments.
 11. Define designing and abstract classes.
- D. Use dynamic method binding, create arrays of subclass objects, use object class and its methods, and demonstrate using the final modifier.
 12. Use dynamic method binding.
 13. Create arrays of subclass objects.
 14. Use object class and its methods.
 15. Demonstrate using the final modifier.
- E. Explain interfaces, the idea behind exceptions, types of exceptions, demonstrate method overloading and multiple constructors, and show an understanding of stream input and output.

16. Explain interfaces.
17. Explain the idea behind exceptions.
18. Explain types of exceptions and dealing with them.
19. Show an understanding of stream input and output.
20. Demonstrate method overloading and multiple constructors.

ASSESSMENT OF COURSE LEARNING OUTCOMES AND COMPETENCIES

Student progress is evaluated through both formative and summative assessment methods. Specific details may be found in the instructor's course information document.

COLLEGE POLICIES AND PROCEDURES

Student Handbook

<https://www.kckcc.edu/files/docs/student-resources/student-handbook-and-code-of-conduct.pdf>

College Catalog

<https://www.kckcc.edu/academics/catalog/index.html>

College Policies and Statements

<https://www.kckcc.edu/about/policies-statements/index.html>

Accessibility and Accommodations

<https://www.kckcc.edu/academics/resources/student-accessibility-support-services/index.html>.