

Washtenaw Community College Comprehensive Report

CPS 261 Advanced Java Concepts Effective Term: Winter 2024

Course Cover

College: Business and Computer Technologies
Division: Business and Computer Technologies
Department: Computer Science & Information Technology
Discipline: Computer Science
Course Number: 261
Org Number: 13400
Full Course Title: Advanced Java Concepts
Transcript Title: Advanced Java Concepts
Is Consultation with other department(s) required: No
Publish in the Following: College Catalog , Time Schedule , Web Page
Reason for Submission: Course Change
Change Information:
 Course description
 Pre-requisite, co-requisite, or enrollment restrictions
 Outcomes/Assessment
 Objectives/Evaluation

Rationale: The course materials are out of date.

Proposed Start Semester: Fall 2023

Course Description: In this course, students will continue exploring Java concepts. Topics covered include input/output, abstract class and Interfaces, graphical user interface (GUI) associated with JavaFX, data structures, Java Stream, multitasking (Threads) and JUnit. Students entering this class should have a good understanding of object-oriented programming concepts such as inheritance and polymorphism. This course is the second part of a two-course sequence.

Course Credit Hours

Variable hours: No

Credits: 4

Lecture Hours: Instructor: 60 Student: 60

Lab: Instructor: 0 Student: 0

Clinical: Instructor: 0 Student: 0

Total Contact Hours: Instructor: 60 Student: 60

Repeatable for Credit: NO

Grading Methods: Letter Grades

Audit

Are lectures, labs, or clinicals offered as separate sections?: NO (same sections)

College-Level Reading and Writing

College-level Reading & Writing

College-Level Math

Level 4

Requisites

Prerequisite

CPS 161 minimum grade "B-"

General Education

General Education Area 7 - Computer and Information Literacy

Assoc in Arts - Comp Lit

Assoc in Applied Sci - Comp Lit

Assoc in Science - Comp Lit

Request Course Transfer

Proposed For:

Student Learning Outcomes

1. Identify input/output techniques including the associated exception handling required.

Assessment 1

Assessment Tool: Outcome-related multiple-choice questions on a departmental exam

Assessment Date: Fall 2023

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Answer key

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

2. Identify graphical user interface (GUI) techniques needed to provide a good program interface for a user.

Assessment 1

Assessment Tool: Outcome-related multiple-choice questions on a departmental exam

Assessment Date: Fall 2023

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Answer key.

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

3. Identify Java data structures and algorithms necessary for efficient programs such as Linked Lists, Stacks, Queues, and Trees.

Assessment 1

Assessment Tool: Outcome-related multiple-choice questions on a departmental exam

Assessment Date: Fall 2023

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Answer key

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

4. Develop a Java program with graphic user interfaces that utilize OOP concepts, Java I/O, JavaFX, Java Stream, Java data structures.

Assessment 1

Assessment Tool: Outcome-related programming exercise

Assessment Date: Fall 2023

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Departmentally-developed rubric

Standard of success to be used for this assessment: 70% of the students will score 70% or higher on the exercise.

Who will score and analyze the data: Departmental faculty

Course Objectives

1. Declare exceptions in a method header and throw exceptions in a method body.
2. Write a try-catch block to handle exceptions and use finally.
3. Read data from a file using the Scanner class and write data to a file using the PrintWriter class.
4. Store and restore objects using ObjectOutputStream and ObjectInputStream.
5. Define interfaces and define classes that implement interfaces.
6. Identify default, static and functional interface.
7. Write a simple JavaFX program and understand the relationship among stages, scenes, and nodes.
8. Create user interfaces using panes, groups, UI controls, and shapes.
9. Layout nodes using Pane, StackPane, FlowPane, GridPane, BorderPane, HBox, and VBox.
10. Describe events, event sources, and event classes.
11. Use Java data structure classes such as stacks, queues, trees, hashing and iterators.
12. Analyze the complexity of algorithms in using the standard Java data structures such as stacks, queues, trees and hashing.
13. Implement the Serializable interface to make objects serializable.
14. Use and define generic classes and interfaces.
15. Explore the relationship between interfaces and classes in the Java Collections Framework hierarchy.
16. Use the common methods defined in the Collection interface for operating collections.
17. Explore how and when to use HashSet, LinkedHashSet, or TreeSet to store a set of elements.
18. Store unordered, nonduplicate elements using a set.
19. Use aggregate operations on collection streams to simplify coding and improve performance.
20. Create a stream pipeline, apply lazy intermediate methods (skip, limit, filter, distinct, sorted, map, and mapToInt), and terminal methods (count, sum, average, max, min, forEach, findFirst, firstAny, anyMatch, allMatch, noneMatch, and toArray) on a stream.
21. Identify what JUnit is and how JUnit works.
22. Create and run a JUnit test class from Eclipse.

New Resources for Course

Course Textbooks/Resources

Textbooks
Manuals
Periodicals
Software

Equipment/Facilities

Level III classroom
Data projector/computer

Reviewer

Action

Date

Faculty Preparer:

<i>Jai Bai</i>	<i>Faculty Preparer</i>	<i>Feb 10, 2023</i>
Department Chair/Area Director:		
<i>Scott Shaper</i>	<i>Recommend Approval</i>	<i>Mar 13, 2023</i>
Dean:		
<i>Eva Samulski</i>	<i>Recommend Approval</i>	<i>Mar 16, 2023</i>
Curriculum Committee Chair:		
<i>Randy Van Wagnen</i>	<i>Recommend Approval</i>	<i>Apr 07, 2023</i>
Assessment Committee Chair:		
<i>Shawn Deron</i>	<i>Recommend Approval</i>	<i>Apr 13, 2023</i>
Vice President for Instruction:		
<i>Victor Vega</i>	<i>Approve</i>	<i>Apr 17, 2023</i>

Washtenaw Community College Comprehensive Report

CPS 261 Advanced Java Concepts

Effective Term: Fall 2015

Course Cover

Division: Business and Computer Technologies

Department: Computer Instruction

Discipline: Computer Science

Course Number: 261

Org Number: 13400

Full Course Title: Advanced Java Concepts

Transcript Title: Advanced Java Concepts

Is Consultation with other department(s) required: No

Publish in the Following: College Catalog , Time Schedule , Web Page

Reason for Submission: Course Change

Change Information:

Consultation with all departments affected by this course is required.

Course description

Pre-requisite, co-requisite, or enrollment restrictions

Outcomes/Assessment

Rationale: Updating prerequisite

Proposed Start Semester: Fall 2015

Course Description: This course is a continuation of the Java concepts covered in CPS 161. Topics covered include input/output, graphical user interfaces associated with AWT/Swing, data structures, networking, and multitasking (Threads). Students entering this class should have a good understanding of object-oriented programming concepts such as inheritance and polymorphism. The title of this course was previously Programming in Data Structures in Java.

Course Credit Hours

Variable hours: No

Credits: 4

Lecture Hours: Instructor: 60 Student: 60

Lab: Instructor: 0 Student: 0

Clinical: Instructor: 0 Student: 0

Total Contact Hours: Instructor: 60 Student: 60

Repeatable for Credit: NO

Grading Methods: Letter Grades

Audit

Are lectures, labs, or clinicals offered as separate sections?: NO (same sections)

College-Level Reading and Writing

College-level Reading & Writing

College-Level Math

Requisites

Prerequisite

CPS 161 minimum grade "B-"

General Education

General Education Area 7 - Computer and Information Literacy

Assoc in Arts - Comp Lit
Assoc in Applied Sci - Comp Lit
Assoc in Science - Comp Lit

Request Course Transfer

Proposed For:

Eastern Michigan University

Student Learning Outcomes

1. Identify input/output techniques including the associated exception handling required.

Assessment 1

Assessment Tool: Multiple choice and short answer questions on a departmental exam

Assessment Date: Fall 2016

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Departmental exam with fixed answers (multiple choice, fill in the blank, etc.).

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

2. Identify graphical user interface (GUI) techniques needed to provide a good program interface for a user.

Assessment 1

Assessment Tool: Multiple choice and short answer questions on a departmental exam

Assessment Date: Fall 2016

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Departmental exam with fixed answers (multiple choice, fill in the blank, etc.).

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

3. Identify multitasking and socket programming techniques necessary to work in an internet driven world.

Assessment 1

Assessment Tool: Multiple choice and short answer questions on a departmental exam

Assessment Date: Fall 2016

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Departmental exam with fixed answers (multiple choice, fill in the blank, etc.).

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

4. Identify Java data structures and algorithms necessary for efficient programs such as trees, hashing and stacks.

Assessment 1

Assessment Tool: Multiple choice and short answer questions on a departmental exam

Assessment Date: Fall 2016

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: All

How the assessment will be scored: Departmental exam with fixed answers (multiple choice, fill in the blank, etc.)

Standard of success to be used for this assessment: 70% of the students who take the exam will score 70% or better.

Who will score and analyze the data: Departmental faculty

5. Develop Java program with graphic user interfaces that include AWT/Swing, networking functions and multitasking.

Assessment 1

Assessment Tool: Programming Exercise

Assessment Date: Fall 2016

Assessment Cycle: Every Three Years

Course section(s)/other population: All

Number students to be assessed: Random sample of 25% of all students with a minimum of one full section

How the assessment will be scored: Departmentally-developed rubric

Standard of success to be used for this assessment: 70% of the students will successfully complete the exercise

Who will score and analyze the data: Departmental faculty

Course Objectives

1. Demonstrate exception handling basics.

Matched Outcomes

2. Demonstrate processing of text and binary files.

Matched Outcomes

3. Use Java data structure classes such as stacks, queues, trees, hashing and iterators.

Matched Outcomes

4. Analyze the complexity of algorithms in using the standard Java data structures such as stacks, queues, trees and hashing.

Matched Outcomes

5. Create a good user interface using Java GUI classes.

Matched Outcomes

6. Use Java multitasking and socket programming techniques.

Matched Outcomes

New Resources for Course

Course Textbooks/Resources

Textbooks

Savitch. *Absolute Java*, ed. Addison-Wesley, 2005

Manuals

Periodicals

Software

Equipment/Facilities

Level III classroom

Data projector/computer

Reviewer

Faculty Preparer:

Clem. Hasselbach

Action

Faculty Preparer

Date

Feb 19, 2015

Department Chair/Area Director:

John Trame

Recommend Approval

Feb 20, 2015

Dean:

Kimberly Hurns

Recommend Approval

Feb 25, 2015

Vice President for Instruction:

Bill Abernethy

Approve

Mar 26, 2015