CS 17.11 Course Outline as of Fall 2018

CATALOG INFORMATION

Dept and Nbr: CS 17.11 Title: JAVA PROGRAMMING
Full Title: Java Programming
Last Reviewed: 1/24/2022

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	3.00	Lecture Scheduled	3.00	17.5	Lecture Scheduled	52.50
Minimum	3.00	Lab Scheduled	0	4	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	3.00		Contact Total	52.50
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00 Total Student Learning Hours: 157.50

Title 5 Category:	AA Degree Applicable
Grading:	Grade Only
Repeatability:	00 - Two Repeats if Grade was D, F, NC, or NP
Also Listed As:	
Formerly:	CIS 17

Catalog Description:
Intended for students with previous programming experience. Topics include: Object-oriented programming principles, Java language constructs, the JDK (Java Developer's Kit), class libraries, Web Services, GUI (Graphical User Interface) development, applications, Java interface to databases.

Prerequisites/Corequisites:
Course Completion of CS 10A

Recommended Preparation:
Eligibility for ENGL 100 or ESL 100

Limits on Enrollment:

Schedule of Classes Information:
Description: Intended for students with previous programming experience. Topics include: Object-oriented programming principles, Java language constructs, the JDK (Java Developer's Kit), class libraries, Web Services, GUI (Graphical User Interface) development, applications, Java interface to databases. (Grade Only)
Prerequisites/Corequisites: Course Completion of CS 10A

Recommended: Eligibility for ENGL 100 or ESL 100
Limits on Enrollment:
Transfer Credit: CSU;UC.
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

AS Degree:	**Area**	Effective:	Inactive:
CSU GE:	**Transfer Area**	Effective:	Inactive:
IGETC:	**Transfer Area**	Effective:	Inactive:

CSU Transfer:	Transferable	Effective:	Summer 1996	Inactive:
UC Transfer:	Transferable	Effective:	Spring 2000	Inactive:

CID:

Certificate/Major Applicable:
Both Certificate and Major Applicable

COURSE CONTENT

Student Learning Outcomes:
At the conclusion of this course, the student should be able to:
1. Use principles of software design to analyze programming problems and develop solutions using the Java programming language
2. Create and test computer programs in the Java programming language that incorporate control structures and object oriented programming methods

Objectives:
Students will:
1. Evaluate the rationale of the Java language
2. Design programs using object-oriented methods
3. Create software using an integrated development environment
4. Integrate the Java class libraries with the construction of new classes
5. Construct graphical user interfaces
6. Investigate web programming using SOAP (Simple Object Access Protocol) and JavaScript Object Notation (JSON)

Topics and Scope:

1. Object-oriented programming principles
   A. Encapsulation
   B. Inheritance
   C. Polymorphism
   D. Aggregation
2. Object oriented design tools/techniques
   A. Unified Modeling Language (UML)
   B. Requirement capture tools
3. Java language constructs

A. Types
B. Operators
C. Flow control
D. Classes
E. Packages and interfaces
F. Intergrated development environment

4. Java Developer Kit (JDK) class libraries
A. .lang
B. .io
C. .util
D. .net

5. Graphical User Interface (GUI) development
A. JavaFX GUI design
B. GUI controls

6. Applications
A. Parameters
B. Initialization
C. Input/Output (I/O)

7. Databases
A. Third party library interfaces to Structured Query Language (SQL) databases
B. NoSQL databases

8. Web Services
A. SOAP
B. JSON

Assignment:

1. Reading, approximately 30 pages per week
2. Write, test, and debug 4 – 12 programs using the Java programming language
3. Write program documentation for each program
4. Take 2 – 4 objective examinations including a final exam

Methods of Evaluation/Basis of Grade:

Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

Written program documentation	Writing 10 - 20%

Problem Solving: Assessment tools, other than exams, that demonstrate competence in computational or non-computational problem solving skills.

Writing, testing and debugging programs using the Java programming language	Problem solving 20 - 60%

Skill Demonstrations: All skill-based and physical demonstrations used for assessment purposes including skill performance exams.

None	Skill Demonstrations 0 - 0%

Exams: All forms of formal testing, other than skill performance exams.

2 – 4 objective examinations including a final exam	Exams 20 - 60%

Other: Includes any assessment tools that do not logically fit into the above categories.

None	Other Category 0 - 0%

Representative Textbooks and Materials:
Java How to Program (Early Objects) (10th). Deitel, Paul; Deitel Harvey. Prentice Hall: 2014