

Course Code: CSC 140 (IAI CS 911)

Course Title: Computer Science I (Java)

Department: Business/Computer Science and Technologies

Effective Date: Summer 2026

PCS Code: 1.1 - Baccalaureate/Transfer

CIP Code: 11.0202

Repeatability: 0

Credit Hours

Catalog Notation: 2-2-3

Credit Hour Distribution:

Lecture: 2

Lab: 2

Clinical: 0

Total: 3

General Course Information

Catalog Description

Introduction to computer science and programming using the Java language. Emphasis on problem solving, algorithm design, and program development including data representation, programming constructs, and object-oriented design fundamentals.

General Course Objectives

- To become proficient in the implementation of Computer Science I level algorithms using the Java programming language.
- To understand and use the basic computer science topics of expressions, selection, iteration, functions, arrays, sorting and searching, and classes.

Minimum Placement Levels

English	Reading	Math
None	None	Placement out of MAT 098

Prerequisites

Credit in CIS 122 or approval of department chair

Methods of Evaluation

10-12 lab programming assignments, 7-8 quizzes, final exam.

Instructional Materials and Additional Supplies

Java Software Solutions, Foundations of Program Design, 9th edition by Lewis and Loftus. ISBN 0134462025

Course Content

General Learning Outcomes (GLOs)

- Reasoning and Inquiry: Students will demonstrate the ability to solve problems using deductive reasoning and logic, quantitative reasoning, or the scientific method.
- Technology: Students will demonstrate the ability to evaluate, select, and appropriately use current and emerging tools.

Course Segments and Student Learning Outcomes

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Course Introduction	1. Identify course requirements, goals, expectations, and course information.	1	0	0
Introduction to Computer Systems and Development Environments	1. Enter, compile, and run a simple program, and identify the major components of programs. 2. Demonstrate installation and operation of the integrated development environment (IDE) and its software development components.	2	3	0
Java Programming Language and Software Development	1. Describe the origin of the Java language and explain the basic components and structure of a simple Java program. 2. Apply basic Java language instructions by constructing a simple program in the development environment. 3. Describe the steps involved in program compilation and execution. 4. Identify and describe the basic principles of object-oriented design. 5. Understand and apply appropriate debugging and testing techniques for the language and environment used.	2	3	0
Data and Expressions	1. Describe and demonstrate the use of character strings and their applications. 2. Describe the declaration, use, and assignment of variables. 3. Define the Java primitive data types and their differences from Java class types. 4. Describe the syntax and processing of expressions, data conversion, and user input processing (interactive programs). 5. Apply variable declarations and expressions to solve a problem using input from a user within a program. 6. Input numeric and string values into variables, write arithmetic and string manipulation expressions to process data, and output the results with formatting.	2	2	0
Using Classes and Objects	1. Describe the processes involved in creating objects and the use of object reference variables. 2. Describe commonly-used classes and methods in the standard Java classes (e.g., String, Math, Random, DecimalFormat). 3. Apply object-oriented principles of creating objects and invoking methods of commonly-used classes in a program to solve a problem statement. 4. Describe enumerated data types and wrapper classes and their uses in applications.	4	4	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Conditionals and Loops	1. Define the concept of flow of control and its process through method calls. 2. Describe boolean expressions and their application in basic decision-making using "if" statements. 3. Describe issues pertaining to comparing certain types of primitive data and object types. 4. Describe the repetitive execution of statements using the "while" looping statement. 5. Plan and develop a program solving a problem statement requiring the application of conditionals and loops. 6. Identify the algorithmic need for selection, choose an appropriate selection structure, and successfully code a select structure that implements the algorithm.	3	3	0
More Conditionals and Loops	1. Describe the use of the "switch" statement in conditional processing and the conditional operator. 2. Describe the repetitive execution of statements using the "do" and "for" looping statements. 3. Identify and describe the differences between the various types of looping statements as well as nested loops. 4. Plan and develop a program solving a problem statement that demonstrates the identification and selection among the various types of conditional and looping statements. 5. Identify the algorithmic need for repetition, choose an appropriate repetition structure, including nested loops if appropriate, and successfully code a repetition structure that implements the algorithm.	2	3	0
Writing Classes	1. Describe the structure and content of a class definition in the context of object-oriented principles including encapsulation, visibility modifiers, and common class methods. 2. Describe and define object state using instance variables and data. 3. Describe the structure of a method definition, including parameters and return values. 4. Demonstrate object-oriented principles by solving a problem statement requiring the development of a user-defined class. Demonstrate the instantiation and operation of an object from the user-defined class within the program. 5. Identify class definitions, objects, and intrinsic data types; declare objects and variables. 6. Implement appropriate program design and documentation methods using a good programming style.	2	2	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Objected-oriented Design	1. Describe key issues and steps related to the design of object-oriented software. 2. Describe strategies and techniques for identifying the classes needed within a problem statement. 3. Describe the effect of the "static" modifier on methods and data. 4. Describe common relationships between classes identified in the design process and self-referencing using the "this" reference. 5. Plan and develop a program that demonstrates the development of multiple classes to solve a problem statement. Identify and demonstrate relationships between the classes in the design and operation of the program. 6. Describe the basic concept of a Java Interface and its use in the design and operation of a program. 7. Describe issues related to the design of class methods including method overloading. 8. Identify and then write method definitions within a class, including the constructor methods; identify the need for parameters and be able to code them; and show the limits of the scope of variables, including instance variables and local variables. 9. Code necessary features of object-oriented languages, including overloaded methods, pass-by-reference, pass-by-value, intrinsic data, return values, comments, simple exception catching, and assertions.	4	3	0
Arrays	1. Define and describe the use of arrays for basic data organization, their difference from the Java ArrayList class, and passing arrays as method parameters. 2. Describe issues related to arrays as objects and arrays of objects. Describe the concept and construction of a two-dimensional array. 3. Describe the use of variable length parameter lists as method parameters as well as the use of command-line arguments within programs. 4. Plan and develop a program requiring the design and development of multiple classes and their relationships as well as arrays of objects to solve a problem statement. 5. Store data into arrays or equivalent structures, including multidimensional arrays or equivalent structures, and walk traverse those arrays or equivalent structures to access the data. 6. Store formatted data to an ASCII text file and read formatted data from an ASCII text file into program variables.	2	2	0
Sorting and Searching	1. Describe the fundamental process of sorting items in a list, including the specific sorting algorithms of selection sort and insertion sort. 2. Describe the fundamental process of searching items in a list, including the specific searching algorithms of linear search and binary search. 3. Identify and analyze examples of both sorting and searching algorithms and their uses within a program. 4. Implement a sort and search on a one-dimensional array or equivalent structures.	2	2	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Inheritance	1. Describe the object-oriented principle involving the derivation of new classes from existing ones including the "protected" visibility modifier and the "super" reference. 2. Describe the concept and purpose of method overriding and its difference from overloading. 3. Describe the object-oriented principle of class hierarchies and their applicability in real-world problem statements; identify and describe the concept of an abstract class. 4. Design and develop a program applying the concept of inheritance by writing a user-defined class derived from an existing class and demonstrating its operation within a program.	4	3	0

Total Contact Hours

Lecture Hours	Lab Hours	Clinical Hours
30	30	0