LOS MEDANOS COLLEGE

ARTICULATION AGREEMENT **DATE DRAFTED**: May 2, 2022
VALID ACADEMIC YEARS: FA22-SP24

LMC COURSE: COMSC-051 "Java for Programmers"
HIGH SCHOOL COURSE AP Computer Science A

School: Heritage High School
Address: 101 American Ave., Brentwood, CA 94513

A. **COLLEGE COURSE DESCRIPTION:** This course is an introduction to Java programming which is used by companies of all sizes as the main programming language to develop various applications and projects worldwide. Students will be introduced to the basic skills needed to write, compile, and run simple Java applications. Topics covered are project designing, object-oriented programming, console applications, graphics applications and many other elements of the Java language.

B. **COLLEGE UNITS:** 3
C. **PRE-REQUISITES: NA**

D. **HIGH SCHOOL/ADULT EDUCATION CLASS:** AP Computer Science A
DESCRIPTION: The AP Computer Science A course is considered a college level introductory course in computer science. A large part of the course is built around the development of computer programs or parts of programs that correctly solve a given problem. The course also emphasizes the design issues that make programs understandable, adaptable, and, when appropriate, reusable. At the same time, the development of useful computer programs and program modules is used as a context for introducing other important concepts in computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, and the study of standard algorithms and typical applications. The AP Computer Science A course utilizes Java.

E. **REQUIRED CONTENT FOR ARTICULATION:**

1. INTRODUCTION

1.1 Understand the Social and ethical ramifications of computer use.
1.2 Understand the ACM's Code of Ethics and Professional Conduct
1.3 This chapter helps students understand the activity of programming.
1.4 They will learn about the architecture of computers.
1.5 They will learn about machine code and high-level programming languages.
1.6 They will recognize compile-time and run-time errors and write pseudocode for simple algorithms.

2. USING OBJECTS

2.1 In this section of the course, students will learn about variables and also understand the concepts of classes and objects.
2.2 Students will be able to call methods learn about parameters and return values and also be able to browse the API documentation.
2.3 They will learn to implement test programs.
2.4 They will understand the difference between objects and object references.

3. IMPLEMENTING CLASSES

3.1 In this chapter, students will become familiar with the process of implementing classes.

3.2 They will understand software copyright laws, intellectual property, freeware, and shareware.

3.3 They will be able to implement simple methods.

3.4 They will understand the purpose and use of constructors.

3.5 They will understand how to access instance variables and local variable and to be able to write javadoc comments.

4. FUNDEMENTAL DATA TYPES

4.1 Students will understand integer and floating-point numbers and recognize the limitations of the numeric data types.

4.2 They will become aware of causes for overflow and roundoff errors.

4.3 They will understand the proper use of constants.

4.4 They will write arithmetic expressions in Java.

4.5 They will use the String type to manipulate character strings.

4.6 They will learn how to read program input and produce formatted output.

5. DECISIONS

5.1 Students will be able to implement decisions using if statements.

5.2 They will effectively group statements into blocks.

5.3 The will learn how to compare integers, floating-point numbers, strings, and objects

5.4 They will correctly order decisions in multiple branches and nested branches

5.5 They will program conditions using Boolean operators and variables.

5.6 They will be able to design tests that cover all parts of a program.

6. ITERATION

6.1 Students will be able to program loops with the while and for statements

6.2 They will be able to avoid infinite loops and off-by-one errors

6.3 They will be able to use common loop algorithms.

6.4 They will understand nested loops, be able to implement simulations, and learn to use a debugger to locate errors in a running program.

7. ARRAYS AND ARRAY LISTS

7.1 Students will become familiar with using arrays and array lists.

7.2 They will learn about wrapper classes, auto-boxing, and the enhanced for loop.

7.3 They will study common array algorithms and learn how to use two-dimensional arrays.

7.4 They will understand when to choose array lists and arrays in their programs.

7.5 They will learn to implement partially filled arrays and understand the concept of regression testing.

8. DESIGNING CLASSES

8.1 Students will learn how to choose appropriate classes for a given problem.

8.2 They will understand the concepts of cohesion and coupling.

8.3 They will learn how to minimize the use of side effects.

8.4 They will understand the need for computer reliability and software redundancy in various computing environments.

8.5 They will document the responsibilities of methods and their callers with preconditions and postconditions.

8.6 They will understand static methods and variables.

8.7 They will understand the scope rules for local variable and instance variables.

8.8 They will learn about packages and also learn about unit testing frameworks.

9. INTERFACES AND POLYMORPHISM

9.1 Students will be able to declare and use interface types.

9.2 The will understand the concept of polymorphism.
9.3 They will appreciate how interfaces can be used to decouple classes.
9.4 They will learn how to implement helper classes as inner classes.
9.5 They will implement event listeners in graphical applications.

10. INHERITANCE

10.1 Students will learn about inheritance and how to inherit and override superclass methods.
10.2 They will be able to invoke superclass constructors.
10.3 They will learn about protected and package access control.
10.4 They will understand the common superclass Object and how to override its toString and equals methods.
10.5 They will learn how to use inheritance for customizing user interfaces.

11. I/O AND EXCEPTION HANDLING

11.1 Students will be able to read and write text files.
11.2 They will learn how to throw and catch exceptions.
11.3 They will be able to design their own exception classes.
11.4 They will understand the difference between checked and unchecked exceptions.
11.5 They will know when and where to catch an exception.

12. OBJECT ORIENTED DESIGN

12.1 Students will learn about the software life cycle.
12.2 They will learn how to discover new classes and methods.
12.3 They will learn how to use CRC cards for class discovery.
12.4 They will be able to identify inheritance, aggregation, and dependency relationships between classes.
12.5 They will master the use of UML class diagrams to describe class relationships.
12.6 They will learn how to use object-oriented design to build complex programs.

13. RECURSION

13.1 Students will learn about the technique of recursion.
13.2 They will understand the relationship between recursion and iteration.
13.3 They will analyze problems that are much easier to solve by recursion than by iteration.
13.4 They will learn to “think recursively”
13.5 They will be able to use recursive helper methods.
13.6 They will understand when the use of recursion affects the efficiency of an algorithm.

14. SORTING AND SEARCHING

14.1 Students will once again familiarize themselves with the Documents and materials addressing privacy issues, legal issues, and intellectual property
14.2 Students will study several sorting and searching algorithms.
14.3 They will appreciate that algorithms for the same task can differ widely in performance.
14.4 They will understand the big-Oh notation.
14.5 They will learn how to estimate and compare the performance of algorithms.
14.6 They will learn how to measure the running time of a program.

15. INTRODUCTION TO DATA STRUCTURES

15.1 Students will learn how to use the linked lists provided in the standard library.
15.2 They will be able to use iterators to traverse linked lists and understand the implementation of linked lists.
15.3 They will distinguish between abstract and concrete data types.
15.4 They will know the efficiency of fundamental operations of lists and arrays.
15.5 They will become familiar with the stack and queue data types.

F. REQUIRED COMPETENCIES (PERFORMANCE OBJECTIVES) FOR ARTICULATION:

1. Design and implement solutions to problems by writing, running, and debugging computer programs.
2. Use and implement commonly used algorithms and data structures.
3. Develop and select appropriate algorithms and data structures to solve problems.
4. Code fluently in an object-oriented paradigm using the programming language Java. Students are expected to be familiar with and be able to use standard Java library classes from the AP Java subset.
5. Read and understand a large program consisting of several classes and interacting objects.
6. Recognize the ethical and social implications of computer use.

G. METHODS FOR END OF COURSE ASSESSMENT:

Homework	10%
Programming Projects	10%
Programming Exercises	10%
Final Exam	20%
Tests/Quizzes	50%

H. TEXTBOOKS OR OTHER SUPPORTING MATERIALS

- Horstmann, Cay. Java Concepts, 2018, 7th Edition Hoboken, N.J.: Wiley 2010
- Online curriculum from (College Board MyAP)
- Online curriculum from www.csawesome.org
- Supplemental: "Multiple Choice & Free Response Questions in Preparation for the AP Computer Science A Examination" Leon Schram 10th Edition. 2021

I. PROCEDURES AND/OR CRITERIA FOR COURSE ARTICULATION:

(all of the following must be met)

1. Students **must apply** to Los Medanos College and register in **CATEMA** in order to receive credit recommendations by their high school teacher.
2. Students **must be recommended** for credit by their high/adult education schoolteacher in **CATEMA.** *Teachers recommend credit at their discretion.*
3. Students **must complete** the AP Computer Science A class at Heritage High School with an overall grade of "B" or better.
   High school/Adult Ed. teachers will enter this grade in CATEMA.
4. Students **must earn** a "B" or better on the agreed upon college/high school final exam procedure.
   High school/Adult Ed. teachers will enter this exam grade in CATEMA.
5. Articulated college credit may only be recommended by the high school teacher and received by the student **within the academic year** in which it was earned.
6. Upon completion of the above, the student will receive on his/her LMC and CCCCD (California Community College District) transcript the units of credit for LMC's **COMSC-051** course.
7. College transcripts will reflect the **FINAL EXAM GRADE** earned and will be notated as *Credit by Exam.

ARTICULATION AGREEMENT

DATE DRAFTED: May 2, 2022
VALID ACADEMIC YEARS: FA22-SP24

LMC COURSE: COMSC-051 "Java for Programmers"
HIGH SCHOOL COURSE AP Computer Science A

School: Heritage High School
Address: 101 American Ave., Brentwood, CA 94513

COLLEGE SIGNATURES

Natalie Hannum
Natalie Hannum (May 3, 2022 14:11 PDT)

Natalie Hannum — Date
LMC Vice President of Instruction

Dennis Franco — Date
LMC Interim Dean of Instruction, Computer Science

Louie Giambattista
Louie Giambattista (May 2, 2022 16:07 PDT)

Louie Giambattista — Date
LMC Computer Science Dept. Chair/Faculty

HIGH SCHOOL/ROP/DISTRICT SIGNATURES

Carrie J Wells
Carrie J Wells (May 12, 2022 10:57 PDT)

Carrie Wells — Date
Principal, Heritage High School

Erik Faulkner
Erik Faulkner (May 12, 2022 12:35 PDT)

Erik Faulkner — Date
LUHSD Associate Superintendent Educational Services

Robert J. Pardi
Robert J. Pardi (May 12, 2022 12:22 PDT)

Robert Pardi — Date
Faculty, Heritage High School