LOS MEDANOS COLLEGE

ARTICULATION AGREEMENT **DATE DRAFTED**: June 25, 2020
VALID ACADEMIC YEAR(S): 2020-21 & 2021-22

LMC COURSE: COMSC-122 "Programming Concepts & Methodologies I"
HIGH SCHOOL COURSE: AP Computer Science A
School: Mountain House High School
Address: 1090 S. Central Parkway, Mountain House, CA 95391

A. **COLLEGE COURSE DESCRIPTION:** This course introduces the discipline of computer science with practical hands-on problem solving using a "high-level" computer programming language. The course will include basic syntax and semantics of a "high-level" language, variables, types, expressions, assignment, basic computation, simple I/O, conditional and iterative control structures, functions and parameter passing, structured decomposition, program design, programming style, algorithms and problem solving strategies, overview of programming languages, binding, visibility, scoping, and lifetime management.

B. **UNITS: 3**
C. **PRE-REQUISITES:** NA
D. **REQUIRED CONTENT FOR ARTICULATION:**

Unit 1: Introduction to Java

1.1. Students will learn about the architecture of computers, machine code and the history of high-level programming languages, and the differences between procedural and object oriented languages.
1.2. They will recognize compile-time and run-time errors and write pseudocode for simple algorithms.1.3. Students will use object oriented design to p
1.4. Students will learn program development using SCRUM.
1.5. Students will u se and understand the difference between the system.out.print and system.out.println commands to print text.
1.6. Students will understand the purpose of and use escape characters and Comments in their code.
1.7. Students will write programs requiring user input and also use string concatenation to format output by creating strings from multiple parts.
1.8. Students will know how to declare a variable, understand what happens in memory when a variable is declared, and use it to store values.,
1.9. Students will write programs with variables of the int, double and char data types and understand the importance of matching inputs to the correct data type.1.10. Students will perform calculations with int and double values, understand how integer
division works in Java, and how to get a more precise value for a division if desired.
1.11. Students will understand the proper use of constants.
1.12. Students will perform casts between int and double data types in both directions and understand the difference between a widening conversion and a narrowing conversion1.13. Students will use modular division in Java to solve problems.
1.14. Students will learn when and how to use the + operator with strings and numerics.
1.15. Students will explore the Java Library, understand how to import and use methods fromthe Math class and others.
1.16. Students will learn what roundoff error means, when/why it can occur, and explore strategies to avoid overflow and round-off errors.
1.17. Students will learn to implement test programs.
1.18. LABS: Movie Ratings, Time Converter

1.19. Quiz and Exam

Unit 2: Conditionals and Loops

2.1. Students will understand the concept of Conditional statements to make choices in a program.
2.2. Students will understand the meaning of the operators ==, <, >, <=, >= and != in Java, and use these to create boolean statements comparing values.
2.3. Students will write programs which use if, else if, and else statements and understand how Java interprets a structure with multiple conditional statements
2.4. Students will understand the meaning of the boolean operators &&, ||, and !, write programs which use boolean operators to create compound boolean statements, and use truth tables to determine when boolean expressions are true or false.
2.5. Students will learn how to use short-circuit evaluation to save time when evaluating boolean statements.
2.6. Students will learn the inverses of the statements x = y, x < y and x > y, and use De Morgan's law to write equivalent boolean statements.
2.7. Students will learn what is meant by the scope and write code using while loops to repeat blocks of commands.
2.8. Students will understand how to use Tracing- how to read and interpret code so that the results of running it can be determined
2.9. Students will understand the purpose of flag variables in code and write code which uses flag variables
2.10. Students will be able to read and write text files.
2.11. Students will learn how and when to throw and catch exceptions, as well as be able to design their own exception classes.
2.12. LABS: Crack the Code!
2.13. Quiz and Exam

Unit 3: Classes, Strings and One-Dimensional Arrays

3.1. They will understand software copyright laws, intellectual property, freeware, and shareware.
3.2. Students will learn the difference between primitive and class data types, the way Java stores these in computer memory, and understand the effect that this has when comparing class data vs comparing primitive data.
3.3. Students will understand what a String is, how it is stored, and write code using several different methods from the String class.
3.4. Students will learn what an array data type is, how to declare and initialize an array, understand how array elements are indexed, and how to extract and edit data which is contained in an array
3.5. Students will learn the syntax of a for loop in Java, and write code which uses for loops to repeat commands, including code which loops through the values in an array.
3.6. Students will understand the meaning of the term "algorithm" and the features which make a good algorithm.
3.7. Students will see examples of common algorithms which are used on arrays and how these are coded in Java.
3.8. Students will understand that arrays can contain class data as well as primitive data, and see examples of code which uses arrays containing the String data type
3.9. Students will understand how numbers can be represented in binary, hexadecimal, and write code that will convert numbers between binary, octal, hexadecimal, and decimal representations.
3.10. LABS: String Shortener, Array Statistics, Merge Arrays
3.11. Quiz and Exam

Unit 4: Methods

4.1. Students will understand what a method is, and learn the syntax for defining and calling void methods and also methods that return values.
4.2. Students will learn to create methods in Java which use parameters.
4.3. Students will understand the difference in the way java treats class type and primitive type parameters and will be able to predict the values of variables after methods are

called with those variables passed as parameters.
4.4. Students will understand the meaning of overloading in Java and will write overloaded methods which have the same name, but take different parameters and may behave differently.
4.5. Students will understand that a method may call itself and that this is called recursion and will be able to Trace code which uses recursion and predict its results.
4.6. Students will learn about the software life cycle.
4.7. They will learn how to use Agile Development process SCRUM
4.8. They will master the use of UML class diagrams to describe class relationships.
4.9. They will learn how to use object-oriented design to build complex programs.
4.10. LABS: Crack the Code
4.11. Quiz and Exam

Unit 5: User-Defined Classes

5.1. Students will understand that an ArrayList is used to hold multiple class type data, are able to write code which declares and adds data to an ArrayList.
5.2. Students will write code that use some of the methods which can be called by objects of the ArrayList class
5.3. Students will write code which implements a for-each loop in Java and understand the advantages and limitations of using a for-each loop versus a standard for loop
5.4. Students will create new public classes in Java with the appropriate constructors, variables, and methods.
5.5. Students will create classes with multiple constructors, learn how to call other methods and constructors from within a class and understand the benefit of doing this.
5.6. Students will understand the difference between static and instance variables and methods, learn some uses of static variables and methods, and understand the modifier FINAL prevents the value of a variable being changed.
5.7. LABS: Fraction, Boxcar & Freight Train, AP Lab: Magpie, AP Lab: Data, AP Lab: Consumer Review
5.8. Quiz and Exam

Unit 6: Advanced Classes

6.1. Students will learn how to create a subclass of an existing class, and understand that public methods are inherited by the subclass of a class and can be accessed from within the class by using the keyword super
6.2. Students will understand that methods inherited by a subclass can be overwritten in the definition for that subclass
6.3. Students will u nderstand the properties of an abstract class and learn how to extend an abstract class with a concrete class which can be instantiated.
6.4. Students will g ain a deeper understanding of complex class hierarchies and inheritance in constructors.
6.5. Students will u nderstand the purpose and behavior of an interface in Java
6.6. Students will understand why wrapper classes for primitive data types may be useful
6.7. Students will understand how autoboxing and unboxing makes writing Java code simpler and understand the circumstances in which Java applies autoboxing and unboxing
6.8. LABS: Ultimate Frisbee, Fraction Comparable, AP Lab: Elevens
6.9. Quiz and Exam

Unit 7: Algorithms - Searching and Sorting

7.1. Students will review the definition of an algorithm and understand their use in searching for and sorting data in arrays and other uses.
7.2. Students will r eview the steps of and understand the working process of the Linear Search, Selection Sort, Insertion Sort, and Binary Search, and Merge Sort Algorithms, and understand the advantages and disadvantages of using each.
7.3. They will understand the big-Oh notation.
7.4. They will learn how to estimate and compare the performance of algorithms.
7.5. They will learn how to measure the running time of a program.
7.6. LABS: Game Wheel, Sort Team Directory, AP Lab: Celebrity
7.7. Quiz and Exam

Unit 8: Two-Dimensional Arrays

8.1. Students will understand the concept of the two-dimensional array in Java and how to declare and initialize and access data from a two-dimensional array in Java

8.2. Students will understand how common algorithms for 2-D arrays are implemented, and write a method which implements an algorithm for processing data in a 2-D array

8.3. LABS: Battleship, AP Lab: Picture, AP Lab: Steganography

8.4. Quiz and Exam

Unit 9: Introduction to Data Structures

9.1. Students will learn how to use the linked lists provided in the standard library.

9.2. They will be able to use iterators to traverse linked lists and understand the implementation of linked lists.

9.3. They will distinguish between abstract and concrete data types.

9.4. They will know the efficiency of fundamental operations of lists and arrays.

9.5. They will become familiar with the stack and queue data types.

9.6. Quiz and Exam

E. REQUIRED COMPETENCIES (PERFORMANCE OBJECTIVES) FOR ARTICULATION

Students will be able to:

1. Design and implement computer-based solutions to problems.
2. Use and implement commonly used algorithms and data structures.
3. Develop and select appropriate algorithms and data structures to solve new problems.
4. Write solutions fluently an object-oriented paradigm
5. Write, run, test and debug solutions in the Java programming language
6. Read and understand programs consisting of several classes and interacting objects
7. Read and understand a description of the design and development process
8. Understand the ethical and social implications of computer use.

F. METHODS FOR END OF COURSE ASSESSMENT:

TEACHING STRATEGIES AND PROCEDURES

- Video lectures, in class demonstrations● Daily programming exercises(collaborative), longer coding projects
- (independent and collaborative)
- Regular quizzes and exams.

GRADING CRITERIA

- MAP Assessments
- Programming Exercises and
- Homework - 10%
  Mastery Assessment
- Programming Projects - 20%
- Final Exam 20%
- Tests/Quizzes 50%
- Students participate regularly in moderated discussion forums as well

G. PROCEDURES AND/OR CRITERIA FOR COURSE ARTICULATION:

a. Complete the AP Computer Science "A" course at Mountain House High School with a grade of "B" or better.
b. Receive a "B" or better on the agreed upon college/high school final exam procedure.
c. Be recommended for credit by the high school teacher.
d. Apply for admission at Los Medanos College.
e. Register for CATEMA for electronic recommendation of college credit **within the academic year in which credit was earned.**
f. 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-122 course.
g. College transcripts will reflect the **FINAL EXAM GRADE** earned and will be notated as *Credit by Exam.

**Distance Learning Circumstances:*
Final Exam "Procedure" will still need to be fulfilled whether the high school class meets in person or moves to a distance learning platform. If the high school class moves to an online learning environment, all efforts will be made to enable students to earn college credit, however due to circumstances beyond the high school/college control, course content may not be able to be completed in order to fulfill the articulation agreement requirements.

H. TEXTBOOKS OR OTHER SUPPORTING MATERIALS

Textbook:

- Eck, David J., Introduction to Programming Using Java, Eighth Edition, July 2019, math.hws.edu/javanotes/index.html
- Learning Management System: Edhesive AP CSA Canvas course
- Software: BlueJ, Dr Java, or another Java editor students are comfortable with (Eclipse or Netbeans)

ARTICULATION AGREEMENT

DATE DRAFTED: June 25, 2020
VALID ACADEMIC YEAR(S): 2020-21 & 2021-22

LMC COURSE: COMSC-122 "Programming Concepts & Methodologies I"
HIGH SCHOOL COURSE: AP Computer Science A
School: Mountain House High School
Address: 1090 S. Central Parkway, Mountain House, CA 95391

COLLEGE SIGNATURES

Natalie Hannum
Natalie Hannum (Jul 7, 2020 14:47 PDT)

Natalie Hannum Date
LMC Vice President of Instruction

Ryan Pedersen
Ryan Pedersen (Jul 7, 2020 11:12 PDT)

Ryan Pedersen Date
LMC Dean of Mathematics and Sciences

Louie Giambattista
Louie Giambattista (Jun 29, 2020 17:28 PDT)

Louie Giambattista Date
LMC Computer Science Department Chair

HIGH SCHOOL/AEC/ROP/DISTRICT SIGNATURES

Ben Fobert
ben fobert (Jul 7, 2020 19:59 PDT)

Ben Fobert Date
Principal, Mountain House High School

Kirk Nicholas
Kirk Nicholas (Aug 24, 2020 14:53 PDT)

Dr. Kirk Nicholas Date
Superintendent, Lammersville Unified School District

Tracey A Lewis
Tracey A Lewis (Jul 8, 2020 12:12 PDT)

Tracey Lewis Date
Faculty, Mountain House High School

MHHS COMSC-122 2020-22 FINAL

Final Audit Report 2020-08-24

Created:	2020-06-29
By:	Colleen Grim (cgrim@losmedanos.edu)
Status:	Signed
Transaction ID:	CBJCHBCAABAA5ikzW5M7jpHaHLHZa1PzBBDmncyQMuki

"MHHS COMSC-122 2020-22 FINAL" History

Document created by Colleen Grim (cgrim@losmedanos.edu)
2020-06-29 - 10:27:36 PM GMT- IP address: 67.169.20.160

Document emailed to Louie Giambattista (lgiambattista@losmedanos.edu) for signature
2020-06-29 - 10:28:23 PM GMT

Email viewed by Louie Giambattista (lgiambattista@losmedanos.edu)
2020-06-30 - 0:26:30 AM GMT- IP address: 24.5.67.217

Document e-signed by Louie Giambattista (lgiambattista@losmedanos.edu)
Signature Date: 2020-06-30 - 0:28:17 AM GMT - Time Source: server- IP address: 24.5.67.217

Document emailed to Ryan Pedersen (rpedersen@losmedanos.edu) for signature
2020-06-30 - 0:28:18 AM GMT

Email viewed by Ryan Pedersen (rpedersen@losmedanos.edu)
2020-07-07 - 6:11:43 PM GMT- IP address: 198.27.161.198

Document e-signed by Ryan Pedersen (rpedersen@losmedanos.edu)
Signature Date: 2020-07-07 - 6:12:08 PM GMT - Time Source: server- IP address: 198.27.161.198

Document emailed to Natalie Hannum (nhannum@losmedanos.edu) for signature
2020-07-07 - 6:12:11 PM GMT

Email viewed by Natalie Hannum (nhannum@losmedanos.edu)
2020-07-07 - 9:47:22 PM GMT- IP address: 99.4.124.154

Document e-signed by Natalie Hannum (nhannum@losmedanos.edu)
Signature Date: 2020-07-07 - 9:47:40 PM GMT - Time Source: server- IP address: 99.4.124.154

Document emailed to ben fobert (bfobert@lammersvilleusd.net) for signature
2020-07-07 - 9:47:42 PM GMT

Email viewed by ben fobert (bfobert@lammersvilleusd.net)
2020-07-07 - 9:54:19 PM GMT- IP address: 66.249.84.14

Document e-signed by ben fobert (bfobert@lammersvilleusd.net)
Signature Date: 2020-07-08 - 2:59:00 AM GMT - Time Source: server- IP address: 73.116.148.163

Document emailed to Tracey A Lewis (tlewis@lammersvilleusd.net) for signature
2020-07-08 - 2:59:02 AM GMT

Email viewed by Tracey A Lewis (tlewis@lammersvilleusd.net)
2020-07-08 - 6:39:37 AM GMT- IP address: 66.102.7.62

Document e-signed by Tracey A Lewis (tlewis@lammersvilleusd.net)
Signature Date: 2020-07-08 - 7:12:40 PM GMT - Time Source: server- IP address: 172.124.235.226

Document emailed to Kirk Nicholas (knicholas@lammersvilleusd.net) for signature
2020-07-08 - 7:12:43 PM GMT

Email viewed by Kirk Nicholas (knicholas@lammersvilleusd.net)
2020-07-08 - 7:20:48 PM GMT- IP address: 107.77.212.54

Email viewed by Kirk Nicholas (knicholas@lammersvilleusd.net)
2020-08-24 - 9:52:38 PM GMT- IP address: 204.129.232.187

Document e-signed by Kirk Nicholas (knicholas@lammersvilleusd.net)
Signature Date: 2020-08-24 - 9:53:09 PM GMT - Time Source: server- IP address: 204.129.232.187

Signed document emailed to Louie Giambattista (lgiambattista@losmedanos.edu), Colleen Grim (cgrim@losmedanos.edu), Tracey A Lewis (tlewis@lammersvilleusd.net), Natalie Hannum (nhannum@losmedanos.edu), and 3 more
2020-08-24 - 9:53:09 PM GMT