Computer Science 2

Course Syllabus (2020-2021)

Noelle Castorena
ncastorena@washoeschools.net

Contents

Course Outline is adopted from Course Syllabus AP Computer Science A © Edhesive
Adapted for WCSD Computer Science 2

Course Pupose

Computer Science II introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language. This industry is in need of qualified people to work in the computer science field. There are not enough qualified people available to fill the amount of jobs across the country. These skills relate to real world careers. Students will have opportunities to compete and participate in Skills USA and Future Business Leaders of America activities throughout the year.

Course Outline

The course includes the required content organized into the following units based on the AP Course and Exam Description:

- Unit 1: Primitive Types
- Unit 2: Using Objects
- Unit 3: Boolean Expressions and if Statements
- Unit 4: Iteration
- Unit 5: Writing Classes

GRADING

Assignments are weighted by group:

Group	Weight
Activities	10%
Quizzes and Assignments	30%
Participation	20%
Final Exam	20%
Exams	20%
Total	**100%**

Course Sequencing

Unit 1: Primitive Types

This unit introduces students to the basics of programming in Java, focusing on the use of variables and operators for storing and manipulating primitive data.

Duration

- 6 Weeks

Unit Topics

- User input and output
- Variables
- Data types
- Calculations using int and double values
- Modular division
- Numeric casting

Lessons

- Unit 1: Lesson 1 - Output In Java: *This lesson demonstrates the system.out.print and system.out.println commands to print text (**MOD**).*
- Unit 1: Lesson 2 - User Input and Variables: *In this lesson, students learn how to store data in variables to be accessed again later (**VAR**).*
- Unit 1: Lesson 3 - Data Types: *This lesson describes different primitive data types that can be used to store numbers in Java (**VAR**).*
- Unit 1: Lesson 4 - Number Calculations: *This lesson introduces arithmetic operators and shows how to perform calculations in Java (**CON**).*
- Unit 1: Lesson 5 - Modular Division: *This lesson explains how modular divisions can be calculated in Java (**CON**).*
- Unit 1: Lesson 6 - Numeric Casts: *This lesson introduces casting to convert one data type to another (**CON**).*

This unit’s lessons include the following activities that reinforce the course’s computational thinking practices:

- Students will complete a partial line of code (**Skill 1.B**)
- Students will determine the results of printing an expression concatenating two strings (**Skill 2.A**).
- Students will explain why a given line of code is incorrect (**Skill 5.B**)
- Students will find an alternative expression which gives the same results in a code segment (**Skill 4.C**).
- Students will determine the results of running a section of code with multiple unary and assignment operators (**Skill 2.B**).
- Students will design a program to individually print out the digits of a three-digit number (**Skill 1.A**).
- Students will determine the result of casting a number (**Skill 2.A**).

Assignment

- Assignment 1: Movie Ratings: *Students will write code using arithmetic operators and variable assignment correctly to calculate values required by the assignment specification (**Skill 2.A**). They will need to fix any errors which occur in their code before submitting it (**Skill 4.B**).*

Assessments

- Unit 1 Quiz
- Unit 1 Exam

Unit 2 - Using Objects

This unit introduces the use of classes, methods and objects.

Duration

- 9 Weeks

Unit Topics

- Primitive vs. class data types
- Computer memory and storage
- String functions and concatenation
- Constructors, classes and objects
- Void and non-void methods
- Wrapper classes
- The Math class

Lessons

- Unit 2: Lesson 1 - Strings and Class Types: *Through the specific example of the String data type, this lesson explains how reference data is stored in memory (**VAR**).*
- Unit 2: Lesson 2 - Escape characters and String Concatenation. *This lesson explains how to use escape characters in String literals, and more about concatenating String variables with primitives (**VAR**)*
- Unit 2: Lesson 3 - String Functions: *This lesson introduces methods which can be called on objects of the String data type (**VAR**).*
- Unit 2: Lesson 4 - Classes and Objects: *This lesson explains the class-object structure found in Java (**MOD**).*
- Unit 2: Lesson 5 - Using Constructors: *In this lesson students are introduced to new classes they can import, and use constructors for creating objects of these class types (**MOD**).*
- Unit 2: Lesson 6 - Using Methods: *This lesson explains how to call methods on objects in Java, and also introduces API documentation which is used to determine which methods can be used for a variable of a given class (**MOD, VAR**).*
- Unit 2: Lesson 7 - Wrapper Classes: *This lesson discusses wrapper classes and explains the processes of autoboxing and unboxing to convert between wrapper class-type variables and primitives (**VAR**).*
- Unit 2: Lesson 8 - Math Functions: *This lesson introduces the static methods from the Math class, used for further mathematical calculations in Java code (**CON**).*

This unit’s lessons include the following activities that reinforce the course’s computational thinking practices:

- Students will determine the output of a statement with multiple different operators used (**Skill 2.A**).
- Students will determine the result of executing a line of code containing multiple method calls (**Skill 2.C**).
- Students will use methods to take user input, then create objects based upon the input (**Skill 3.A**).
- Students will choose a code segment to change the state of an object which is declared in the question text (**Skill 1.C**)

- Students will write code which creates objects, then call methods to both determine and change the state of the object (**Skill 3.A**).
- Students will explain why a line of code is incorrect (**Skill 5.B**).

Assignment

- Assignment 2: Control Tower: *Students will use the constructors of a new class type to create objects, and call methods on the objects created to both change and access their states (**Skill 3.A**).*

Assessments

- Unit 2 Quiz
- Unit 2 Exam

Unit 3 - Boolean Expressions and If Statements

This unit introduces the concepts that programs use to make decisions using the logic of Boolean expressions, If and Else statements.

Duration

- 8 Weeks

Unit Topics

- If and else statements
- Booleans and truth tables
- Short circuit evaluation and De Morgan's law
- Comparing objects

Lessons

- Unit 3: Lesson 1 - Simple Ifs: *In this lesson, students are shown how to use if statements to introduce conditional logic to their program (**CON**).*
- Unit 3: Lesson 2 - Ifs - Making Decisions: *This lesson introduces more comparison operators which can be used in if statements with primitive values (**CON**).*
- Unit 3: Lesson 3 – Else: *This lesson introduces the else and else if statements to give students greater control over the flow of their programs (**CON**).*
- Unit 3: Lesson 4 - Booleans and Truth Tables: *This lesson introduces the use of Boolean operators to create compound Boolean statements (**CON**).*
- Unit 3: Lesson 5 - Short Circuit Evaluation: *This lesson explains how short-circuit evaluation is used by Java when evaluating Boolean statements (**CON**).*
- Unit 3: Lesson 6 - De Morgan's Law: *In this lesson students learn how to apply De Morgan's law to find equivalent Boolean statements (**CON**).*
- Unit 3: Lesson 7 - Comparing Objects: *This lesson explains how to compare object data in Java using both the '==' operator and the equals method (**CON**).*

This unit's lessons include the following activities that reinforce the course's computational thinking practices:

- Students will identify an error in a short segment of code (**Skill 4.B**).
- Students will identify which code segments from a selection are equivalent (**Skill 4.C**).
- Students will complete a partially completed code segment (**Skill 1.B**).
- Students will write code with multiple conditional statements to produce a desired output (**Skill 3.C**). ● Students will describe the details of how a particular statement will be evaluated in a Java program (**Skill 5.A**).
- Students will identify equivalent Boolean statements (**Skill 4.C**).
- Students will evaluate the output from a code segment containing multiple conditionals (**Skill 2.B**).

Assignment

- Assignment 3 - Crack the Code!: *Students will write nested decision statements to control the flow of a program, determining which conditions are checked at which points in the program to produce the correct output (**Skill 2.B**).*

Assessments

- Unit 3 Quiz
- Unit 3 Exam

Unit 4 - Iteration

This unit introduces the idea of *iteration*, repeating a process until certain conditions are met. This unit focuses on while and for loops to create algorithms for Strings and numbers. Unit 4 also contains the first College Board lab assignment.

Duration

- 6 Weeks

Topics

- Algorithms for numbers and Strings
- While, for and nested loops

Lessons

- Unit 4: Lesson 1 - While Loops: *In this lesson, students are introduced to the use of while loops to repeat program code (**CON**).*
- Unit 4: Lesson 1 1/2 - Tracing Code: *This lesson demonstrates how the results of code segments containing iterative statements can be determined by using a trace table (**CON**).*
- Unit 4: Lesson 2 - Algorithms for Numbers: *This lesson introduces standard algorithms which students can code, modify and develop using loops (**CON**).*
- Unit 4: Lesson 3 - The For Loop: *This lesson introduces the for loop, an alternative loop syntax used to repeat code (**CON**).*
- Unit 4: Lesson 4 - Algorithms for Strings: *This lesson focuses on the creation, implementation and modification of algorithms which use String traversals (**CON**).*
- Unit 4: Lesson 5 - Nested Loops: *In this lesson, the way in which nested iteration statements can be used is examined (**CON**).*
- Unit 4: Lesson 6 - Algorithm Efficiency: *This lesson introduces the concept of algorithmic efficiency through the use of statement execution counts (**CON**).*

This unit’s lessons include the following activities that reinforce the course’s computational thinking practices:

- Students will explain why a given code segment will not run as intended (**Skill 5.B**).
- Students will explain what is calculated by a given code segment (**Skill 5.A**).
- Students will complete missing code in a loop header (**Skill 1.B**).
- Students will use both selection and iteration to produce a desired output when the main method of a program executes (**Skill 3.C**).
- Students will explain the results of running a code segment (**Skill 5.A**).
- Students will determine the results of running a code segment with nested iteration statements (**Skill 2.B**).
- Students will write nested iteration statements to produce a desired output (**Skill 3.C**).

Assignments

- Assignment 4: String Shortener: *Students will design a program which processes a String, determines properties of this String and prints a modified version of the String (**Skill 1.A**). Students must write iteration statements, decision statements, and String methods to implement this program (**Skill 2.B**).*

Assessments

- Unit 4 Quiz
- Unit 4 Exam

Unit 5 - Writing Classes

This unit introduces the concepts required for students to write their own classes, which will form the basis for creating more sophisticated programs later in the course. This unit also contains the first of two lessons examining the wider impacts of computer science on society and culture.

Duration

- 6 Weeks

Topics

- Void and Return methods
- Parameters
- Classes
- Constructors
- Static vs. Instance
- Wider Impacts of Computing **Lessons**
- Unit 5: Lesson 1 - Void methods: *This lesson explains how code can be written in methods which can then be called from the main code (**MOD**).*
- Unit 5: Lesson 2 - Parameters: *In this lesson, students are shown how to write methods which use parameters to send information with a method call (**MOD**).*
- Unit 5: Lesson 3 - Parameters - Primitive vs. Class: *This lesson explains how parameters of primitive and class-type data are passed to methods when they are called (**MOD**).*
- Unit 5: Lesson 4 - Return Methods: *This lesson introduces the use of return statements to allow methods to send information back when they are called (**MOD**).*
- Unit 5: Lesson 5 - Classes - The Basics: *Through this lesson, students are introduced to the basics of custom class creation, including writing constructors and instance methods/variables (**MOD**).*
- Unit 5: Lesson 6 - Constructors: *This lesson shows in greater depth how constructors can be written to set the initial state of an object, and explains how formal parameters and local variables differ from global variables in scope (**MOD, VAR**).*
- Unit 5: Lesson 7 - Documenting a Class: *In this lesson students learn how to write comments to document a class, including using preconditions and postconditions (**MOD**).*
- Unit 5: Lesson 8 - Static Vs. Instance: *This lesson covers the static modifier which can be used to associate variables or methods to a class rather than with an object (**MOD**).*
- Unit 5: Lesson 9 - Wider Impacts of Computing: *For this lesson, students must conduct research into aspects of the ethical and social implications of computing systems (**IOC**).*

This unit's lessons include the following activities that reinforce the course's computational thinking practices:

- Students will determine what is printed by a method that uses selection (**Skill 2.B**).
- Students will write a method which uses selection to meet a specification for what will be printed (**Skill 3.C**).
- Students will determine what is output by a code segment which calls a method (**Skill 2.C**).

- Students will write a method using iteration and selection which returns values according to a specification (**Skill 3.C**).
- Students will create a custom class, which is used to create objects representing vehicles (**Skill 3.B**).

Assignment

- Assignment 5: Fraction: *Students will write the constructors and methods of a class designed to represent fractions (**Skill 3.B**). They will need to write the class methods according to strict specifications as to how the methods should behave for different values of parameters and/or member variables (**Skill 3.C**).*

Assessments

- Unit 5 Quiz
- Unit 5 Exam

Institutional Policies

For more details of the policies at the Academy of Arts, Careers and Technology, or the Washoe County School District view the Student / Parent handbook

Mode of Instruction and class meeting time(s) Class meeting will vary based on personal or district schedule

In Person Class meets face to face, on A days for 85 minutes in room 334.

Hybrid program will meet in person and remotely. Students will be expected to come to class virtually on remote days through Teams.

Distance learning will be conducted entirely through Edhesive, Microsoft Teams, and Remind. Students will use Teams to participate virtually in the in person class.

Special Accommodations

In accordance with the American with Disabilities Act and section 504 of the Rehabilitation Act of 1973, students with a documented disability are eligible for support services and accommodations. Services for students with disabilities are offered through the WCSD. Possible accommodations for disabilities include extended testing time, test- taking in isolation, computer use for test taking, tape recorders in class, study skills counseling and shared notetaking in classes. If a student wishes to request an accommodation in one of his or her classes, the student may call WCSD

Attendance, Participation, and Attitude

Attendance, participation, and a positive attitude are required in this class. In the work world, you are not allowed to be late to work and since we are developing your work skills, tardiness may not be tolerated. If you are going to be late for class, you may email or send a message on remind to the teacher in advance to notify her. Attendance in all classes is expected; However, just as in the work world absenteeism occurs; the difference is, you are not paid when you do not work. If you are going to miss a class, you may notify the teacher, in advance, to notify them. Your grade could be affected by not being in the lab finishing projects.

Academic Honesty/ Plagiarism Policy

AACT's Honest/ Plagiarism Policy may be adhered to in this class. Students in this course, as well as all courses, should be aware of strong sanctions against plagiarism as stated in the current Student Handbook. Plagiarism may result in an automatic "F" in the course work and possible expulsion from AACT. In this context, forms of academic dishonesty include but are not limited to:

1. Cheating on tests, examinations, case studies, and other class work.

2. Involvement in plagiarism (the appropriation of another work and the unacknowledged of that work in one's own written work offered for credit.)

3. Collusion (the unauthorized collaboration with another person in preparing course work.

A written statement of the official policies, procedures, and processes related to Academic Honesty/Plagiarism can be found in the Student / Parent Handbook.