WeTeach_AP®CSA

A Computer Science Curriculum
for the College Board Computer Science Advanced Placement Test

Syllabus

Overview

WeTeach_AP®CSA is a curriculum, completely aligned to the College Board Computer Science "A" test CED (Course Exam Description, Summer 2019 release) designed to support computer science teachers in helping their students prepare for and attempt to sit for the CSA AP test. It is presented to both teachers and students using the Canvas LMS, providing E-lessons using videos, interactive activities, practice questions and exercises, labs (using Repl.it, an online IDE), lab solutions, projects, and assessments (both formative and summative) designed for approximately 120 days of instruction. Teachers have access throughout the year to discussion boards, online professional support, and virtual office hours with the course developers to provide them with the tools they need to master and effectively deliver to their students the curriculum content.

Texts and Resources *(CR1)*

A college level textbook can be chosen from any of the following:

- CompuScholar, Inc. TeenCoder: Java Programming. Alpharetta, GA: http://www.compuscholar.com/.
- Cook, Charles E. Blue Pelican Java. Refugio, Texas: Charles E. Cook, 2010.
- Dean, John, and Raymond Dean. Introduction to Programming with Java: A Problem Solving Approach. McGraw-Hill.
- Ericson, Barbara. Java Review for the AP CS A Exam. Online Text.
- Horstmann, Cay. Big Java: Early Objects. Hoboken, N.J.: Wiley.
- Horstmann, Cay. Java Concepts. Hoboken, N.J.: Wiley.
- Kolling, Michael, and David Barnes. Objects First with Java: A Practical Introduction Using BlueJ. Upper Saddle River, N.J.: Prentice Hall.
- Lambert, Ken, and Martin Osborne. Fundamentals of Java: AP Computer Science Essentials. National Geographic Learning/Cengage Learning.
- Lewis, John, and William Loftus. Java Software Solutions: Foundations of Program Design. Upper Saddle River, N.J.: Prentice Hall.
- Litvin, Maria, and Gary Litvin. Java Methods. Andover, Mass.: Skylight Publishing.
- Reges, Stuart, and Marty Stepp. Building Java Programs. Boston, Mass.: Addison-Wesley.

The IDE chosen for this course is repl.it which allows online coding, submission and grading of programming projects.

Student Practice

Most E-Lessons provide formative assessment as the lesson is presented and all E-Lessons provide practice questions at the end of each topic lesson. Offline summative quizzes will be administered at the end of each topic lesson. There will be a summative unit exam for each of the ten units. Prior to the summative unit exams students will be provided the opportunity to complete a Personal Progress Check for that unit. These checks will be formative in nature.

Lab Requirements

There are 129 programming assignments as well as a programming project for each of the ten units. These assignments and projects will provide well over the 20 hour in-class programming requirement. *(CR9)*

Course Outline *(CR2)*

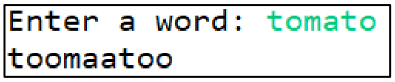
Note: *In the descriptions below, a class period is defined as a 45-minute session. Also, all unit descriptions and time estimates for coverage are taken directly from the CED, and the* WeTeach_AP®CSA *is in complete compliance with the requirements contained within this document. Additional comments in italics are provided by the curriculum author.*

- **Unit 1 – Primitive Types** – 10 class periods, 6 topic lessons – This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units.
- **Unit 2 – Using Objects** – 15 class periods, 6 topic lessons – In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer’s purpose.
- **Unit 3 – Boolean Expressions and *if* Statements** – 13 class periods, 7 topic lessons – Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program.
- **Unit 4 – Iteration** – 16 class periods, 5 topic lessons – This unit focuses on iteration using while and for loops, …and introduces several standard algorithms that use iteration.
- **Unit 5 – Writing Classes** – 14 class periods, 9 topic lessons – This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes, … and focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes.
- **Unit 6 – Array** – 8 class periods, 4 topic lessons – This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. … *It also covers* standard algorithms to use with … arrays. Additional standard algorithms, such as standard searching and sorting algorithms, *will also be introduced*.
- **Unit 7 – ArrayList** – 12 class periods, 6 topic lessons – *This unit covers the ArrayList data structure object defined by Java.* As students learned in Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier.
- **Unit 8 – 2D Array** – 12 class periods, 2 topic lessons – *This unit covers the two-dimensional array data structure.* In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit.
- **Unit 9 – Inheritance** – 15 class periods, 6 topic lessons – *This unit introduces the idea of inheritance within class hierarchy definitions.* One of the strongest advantages of Java is the ability to categorize classes into hierarchies through inheritance. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. *This unit explores the fundamentals of class attributes and behaviors within an inheritance hierarchy structure.*
- **Unit 10 – Recursion** – 5 class periods, 2 topic lessons – Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called **recursion**, …a powerful math and computer science idea, *suitable to use in many computer science algorithms and data structure processing techniques.*

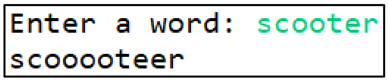
Selected Assignment Descriptions

(CR3, MOD, VAR and CON, CR6, CTP3.D, CR9) Practice Lab 7.4.1*(CR9)* – AddVowels

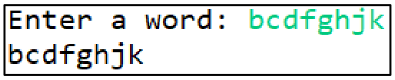
Write a program that will allow the user to enter a single word. Place each individual letter of that word into an ArrayList ***(CR3, VAR)*** that holds String objects ***(CR3, MOD)***. Using a loop of your choice ***(CR3, CON)***, traverse the ArrayList and for each letter in the ArrayList that is a vowel add that same vowel to the list at that location in the list. Then, using a for each loop, traverse the list again and print each of the letters. The output should be a word where every vowel appears twice. Here are three sample runs:



```
Enter a word: tomato
toomaatoo
```



```
Enter a word: scooter
scooooteer
```



```
Enter a word: bcdfghjk
bcdfghjk
```

(CR3, MOD, CR4, CTP1.C, CR6, CTP3.A, CR9) Practice Lab 2.3.1*(CR9)* – RectangleFun

Write a program that will allow the user to enter the width, height, and the coordinates for the location of a rectangle. The program should then calculate and print the perimeter of the rectangle. After printing the perimeter, move the rectangle 4 units to the left and 2 units up and print the new location. The prompts and the labels for the output should appear exactly like those shown in the sample run on the right. You are required to use method calls for all calculations and output. DO NOT use the variables used to gather data from the user. The signatures for the methods needed for this exercise are as follows:

```
public Rectangle(int x,int y,int width,int height)
public double getWidth()
public double getHeight()
public double getX()
public double getY()
public void setLocation(int x,int y)
```

The following starter code is provided for the students ***(CTP1.C, CR4)***:

```
import java.awt.Rectangle;
import java.util.Scanner;
public class RectangleFun
{
        public static void main(String[] args)
        {
                Scanner kb = new Scanner(System.in);
                System.out.print("Width: ");
                int w = kb.nextInt();
                System.out.print("Height: ");
                int h = kb.nextInt();
                System.out.print("X: ");
                int x = kb.nextInt();
                System.out.print("Y: ");
                int y = kb.nextInt();

                // Instantiate a Rectangle object using w, h, x and y. (MOD, CR3, CTP3.A, CR6)
```

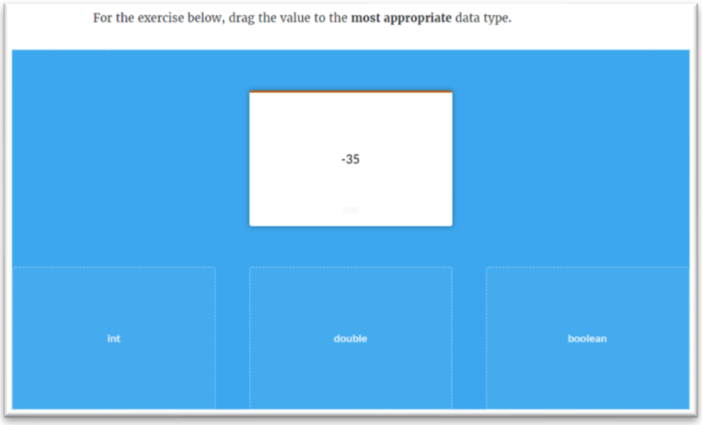
```
                    // Calculate and print the perimeter using Rectangle class methods.

                    // Move the rectangle 2 units up and 4 units to the left using a Rectangle class method.

                    // Print the new location.

          }
}
```

(VAR, CR3) Interactive Sorting Activity 1.2.7 – Data Type Practice

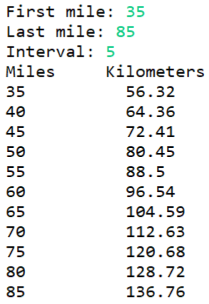

For the exercise below, drag the value to the **most appropriate** data type.

Students are to drag each of the values shown in the top box into one of the boxes below which are labelled as one of three data types ***(VAR, CR3)***.

(CR3, VAR and CON, CR4, CTP1.A, CR5, CTP2.A, CR6, CTP3.C) E-Lesson 4.2.4 - MilesToKilometers

Print a chart that converts miles to kilometers. Allow the user to enter the starting and ending values for the miles and the interval for each entry in the chart. The miles will be shown as whole numbers but the kilometers will be decimal numbers. Round the kilometers to 2 decimal places.

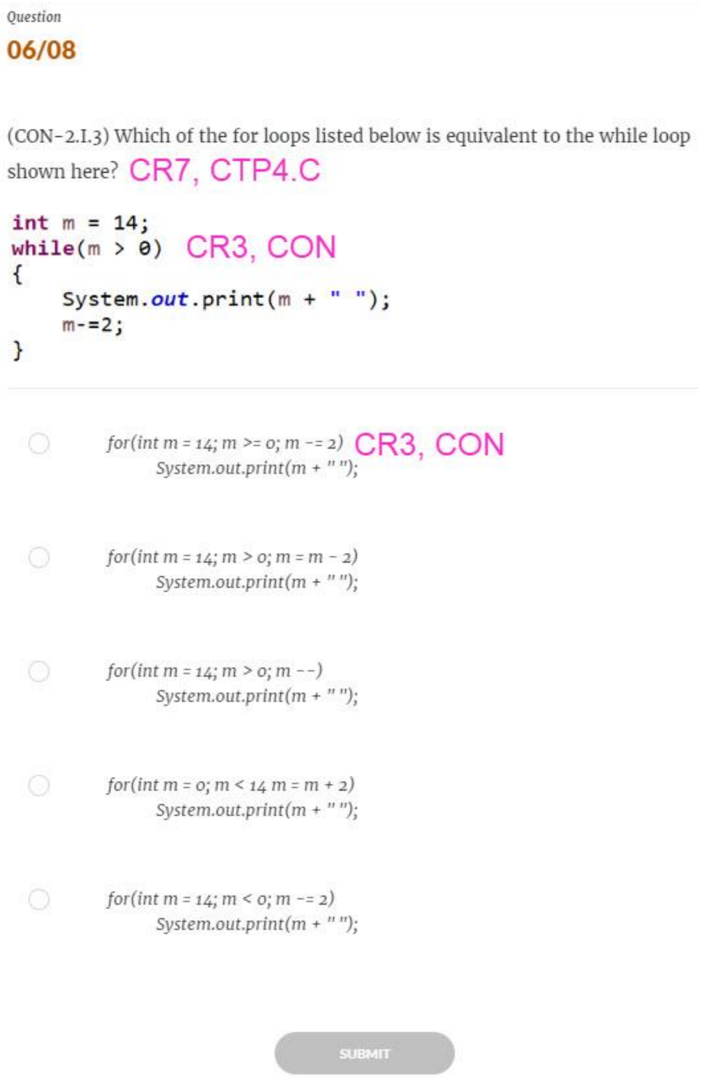
A sample run might look like this:



```
First mile: 35
Last mile: 85
Interval: 5
Miles     Kilometers
35          56.32
40          64.36
45          72.41
50          80.45
55          88.5
60          96.54
65          104.59
70          112.63
75          120.68
80          128.72
85          136.76
```

Students are led through the process of finding the formula to convert miles to kilometers***(CTP1.A, CR4)***, developing an expression that rounds the kilometers to two decimal places***(VAR, CR3, CTP2.A, CR5)***, and writing code to print a chart that makes use of a for loop***(CON, CR3, CTP3.C, CR6)***.

(CR3, CON, CR7, CTP4.C) E-Lesson 4.2.6 - Practice Questions – Question #6



Question
06/08

(CON-2.1.3) Which of the for loops listed below is equivalent to the while loop shown here? CR7, CTP4.C

```
int m = 14;
while(m > 0)  CR3, CON
{
    System.out.print(m + " ");
    m-=2;
}
```

○
```
for(int m = 14; m >= 0; m -= 2)  CR3, CON
        System.out.print(m + " ");
```

○
```
for(int m = 14; m > 0; m = m - 2)
        System.out.print(m + " ");
```

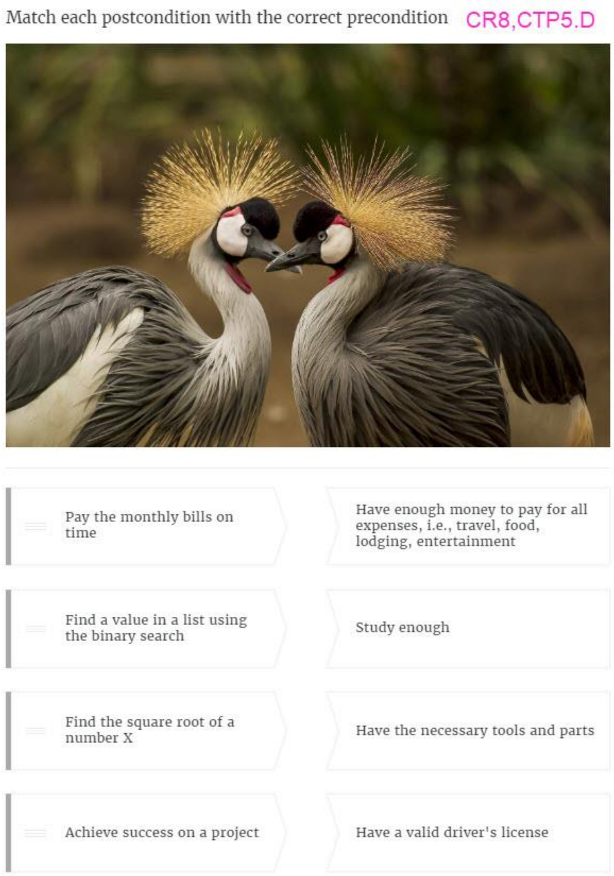
○
```
for(int m = 14; m > 0; m --)
        System.out.print(m + " ");
```

○
```
for(int m = 0; m < 14 m = m + 2)
        System.out.print(m + " ");
```

○
```
for(int m = 14; m < 0; m -= 2)
        System.out.print(m + " ");
```

In the event a student chooses an incorrect answer they are provided with immediate feedback and provided the opportunity to try the question again.

(CR8, CTP5.D) E-Lesson 5.3.6 - Knowledge Check – Matching Exercise



Match each postcondition with the correct precondition CR8,CTP5.D

Pay the monthly bills on time

Have enough money to pay for all expenses, i.e., travel, food, lodging, entertainment

Find a value in a list using the binary search

Study enough

Find the square root of a number X

Have the necessary tools and parts

Achieve success on a project

Have a valid driver's license

There are ten pre and post conditions to be matched in this exercise.

(CR3, IOC) E-Lesson 7.7.2 - Data Collection Privacy Issues – Video

The video explains that computer use and the creation of programs have an impact on personal security and that these impacts can be beneficial and/or harmful. Students learn that when using the computer, personal privacy is at risk and that programmers should attempt to safeguard personal privacy. ***(CR3, IOC)***

Curriculum Features

- **Teacher Resources**
  - Canvas based teacher-facing LMS, including
    - Lesson Plans, fully aligned with CED
    - Lab Solutions
    - Assessments and Keys with Explanations for all topic lessons and units
    - E-Lessons, multi-platform accessible (PCs, tablets, smart phones)
    - Videos, both embedded in E-Lessons and separately accessible through Vimeo, with captions
    - Discussion Board for interacting with other participating teachers
  - Labs on Repl.it (an online IDE, College Board recommended)

- **Student Resources**
  - Canvas based student-facing LMS, including
    - E-Lessons, activities, and videos only
    - No Discussion Board access
  - Repl.it access for lab work at school or home

- **Repl.it online IDE**

- A free and online IDE,
  - Suitable for labs with internet access only,
    - as well as home availability for students without requirement of IDE installation
  - College Board recommended
  - Over 100 labs defined and available
  - Cloned specifically for each school, managed by local teacher

- **Online support**
  - Canvas Discussion Board for teachers, monitored and managed by course developers
  - Periodic Office Hours – for all subscribing teachers
  - Master Classes On Demand – for teachers and students of a school, available on request by appointment