Dominican International School
AP Computer Science A
SY: 2022-23

Grade Level 11/12
1 Year

Teacher Tim Sampson
Email: tsampson@dishs.tp.edu.tw

Course Description

AP Computer Science A is equivalent to a first-semester, college level course in computer science. The course introduces students to computer science with fundamental topics that include problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing. For more details see the college board AP Computer Science A Course and Exam Description—Summer 2020

Requirements

The necessary prerequisites for entering the AP Computer Science A course include knowledge of basic algebra and experience in problem solving. A student in the AP Computer Science A course should be comfortable with functions and the concepts found in the uses of functional notation, such as f(x) 5 x 1 2 and f(x) 5 g(h(x)). It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.

Course Objectives

The course emphasizes both object-oriented and imperative problem solving and design using Java language. These techniques represent proven approaches for developing solutions that can scale up from small, simple problems to large, complex problems. Students cultivate their understanding of coding through analyzing, writing, and testing code as they explore concepts like modularity, variables, and control structures.

Primary Textbook and Other Resources

- Online access to programming labs, homework, e-text and supplemental videos:
  - https://runestone.academy/runestone/books/published/csawesome/index.html
- Our class site for discussion forums, collaboration, notes and assignments. https://classroom.google.com
- Supplementary documents, activities, labs and examples:
  - The Java™ Tutorials
  - AP CSA Labs We will cover the old labs, Chatbot, PixLab as well as the new labs Steganography, Celebrity and Data. They will be introduced as indicated in the schedule below.
- Physical Copy ~ 5 Steps to a 5: AP Computer Science A 2018 2nd Edition by Dean R. Johnson , Carol A. Paymer , Aaron P. Chamberlain 5 Steps to a 5
- Online ~ CSAwesome Runestone Course https://runestone.academy/runestone/books/published/csawesome/index.html
- AP Classroom AP Checks, Unit Reviews and Videos.https://apclassroom.collegeboard.org/

Assessment

Assignments 30%
Quizzes 30%
Quarter Exam 30%

Deportment 10 %

Additional Information:

Lab Component:

The course includes a structured-lab component in which students will complete a minimum of 20 hours of hands-on lab experiences. The curriculum has small coding assignments called Programming Challenges in each lesson, which they are encouraged to do using pair programming. In addition, students will complete at least three of the following College Board AP Computer Science A labs, as chosen by their teachers, in the CSAwesome course to complete a minimum of 20 hours:

- ❏ MagPie Chatbot Lab (built into the CSA curriculum)
- ❏ Picture Lab (built into the CSA curriculum)
- ❏ Consumer Review Lab (built into the CSA curriculum)
- ❏ Celebrity Lab
- ❏ Steganography Lab
- ❏ Data Lab

AP CS A Exam

Students who complete this course will be prepared to take the AP CS A Exam in May.

LTO's D'TORCH (Truthful, Organized, Reflective, Courageous and Helpful)

In CS classes the categories of the D'TORCH most practiced and assessed are:

- Organized - Students utilize Google Classroom to edit, submit and keep track of their assignments.
- Reflective - Students will regularly write activity reflections in their online journal.
- Helpful - Students are empowered to ask for and provide explanations and give examples to help classmates through particularly difficult problems.

Class Expectations

- Come to class on time and be prepared
- Have a positive attitude and be willing to learn.
- Respect yourself, others, and our school.
- Always complete your work and try your best.
- Actively participate, listen carefully, but don't speak out of turn.
- All assignments must be completed.

Homework and Quiz Rules

- All assignments must be turned in on the day they are due.
- 1 day late = Maximum of only 60%
- 2+ days late = Project-I & Only 60%
- If a student has been absent, it is his/her duty to find out what work is due, and hand it in a day later.
- All assignments must satisfactorily be completed.
- If you are absent on the day of a quiz, you will only be able to get a maximum of 60%.

Classroom Rules

- All students are expected to follow the rules. Consequences will follow if rules are broken.
- Read and follow the standard school rules.
- Be on time and neatly dressed, in full school uniform.
- Speak in ENGLISH ONLY.
- Respect your teachers, fellow students and their property.
- Keep your seating space and classroom clean and neat.
- No eating or drinking in the ICT Labs.
- Ask permission to leave the class.
- Neither cheating nor copying in any form will be accepted. Anyone caught doing either during an activity, project or assessment will be given a zero.

Academic Honesty (Plagiarism)

HS CS at DIS is adopting the Harvard CS guidelines on academic honesty. From the Harvard Syllabus
"The essence of all work that you submit to this course must be your own. Collaboration on problems is not permitted (unless explicitly stated otherwise) except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code or writing to others, but you may not view theirs, so long as you and they respect this policy's other constraints."
More details can be found at this link: https://cs50.harvard.edu/ap/2023/syllabus/#academic-honesty

Weekly Schedule

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 **Aug 10th to 12th** **3 Days of Class** *10~ First Day / Orientation Day*	**Unit 1: Getting Started and Primitive Types** 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. Students will focus on writing the main method and will start to call pre-existing methods to produce output. The use of pre-existing methods for input is not prescribed in the course; however, input is a necessary part of any computer science course so teachers will need to determine how they will address this in their classrooms. Through interactive coding challenges and exercises built into the lessons, students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations **(VAR)**. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. Reference data will be covered in Unit 2. 1.1. Getting Started
Week 2 **Aug 15th to 19th** *Opening Mass*	1.2. What is Java? 1.3. Variables and Data Types 1.4. Expressions and Assignment Statements 1.5. Compound Assignment Operators 1.6. Casting Variables 1.7 to end: Summary, Exercises, Review, AP Classroom U1 Quiz
Week 3 **Aug 22nd to 26th**	**Unit 2: Using Objects** 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. Students will learn about modularity in object-oriented programming, which allows us to use abstraction to break complex programs down into individual classes and methods, through interactive coding challenges and exercises built into the lessons **(MOD)**. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships. 2.1. Objects - Instances of Classes 2.2. Creating and Storing Objects: Constructors

	2.3. Calling Methods Without Parameters 2.4. Calling Methods With Parameters 2.5. Calling Methods that Return Values 2.6. Strings
Week 4 **Aug 29th to Sep 2nd**	2.7. String Methods 2.8. Wrapper Classes - Integer and Double 2.9. Using the Math Class 2.10 to end: summary, practice, AP Classroom U2 Quiz
Week 5 **Sep 5th to 9th** **4 Days of Class** *8~ Mass &Birthday Mother Mary* *9th – Moon Festival*	**Unit 3: Boolean Expressions and If Statements** 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. Students will learn that conditional control structures give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program **(CON)**. Through interactive coding challenges and exercises as well as the MagPie Chatbot Lab, students learn that selection and iteration work together to solve problems. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators. The third building block of all algorithms is iteration, which will be covered in Unit 4. 3.1. Boolean Expressions 3.2. if Statements and Control Flow 3.3. Two-way Selection: if-else Statements 3.4. Multi-Selection: else-if Statements
Week 6 **Sep 12th to 16th** FYI – Pre-Exam Days	3.5. Compound Boolean Expressions 3.6. Equivalent Boolean Expressions (DeMorgan's Laws)
Week 7 **Sep 19th to 23rd**	3.7. Comparing Objects 3.8. Summary and Practice 3.9. Magpie Chatbot Lab
Week 8 **Sep 26th to 30th** **2 Days of Class** *28-30 ~Teacher's Conference*	3.10 to end: summary, exercises, AP Classroom Unit Quiz Final Exam U1-U3
Week 9 **Oct 3rd to 7th** **3 Days of Class** *6-7 ~Q1 Exams*	**Review / Semester 1 Project**

2nd QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 (10) **Oct 10th to 14th** **4 Days of Class** *10 – Double 10 Holiday*	**Unit 4: Iteration (Loops)** This unit focuses on iteration using while and for loops. Students will have learned that boolean expressions are useful when a program needs to perform different operations under different conditions. In this unit, they will learn that boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations.

	Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. Students will be able to determine the number of times that a code segment will execute by doing a run-time analysis and using a code tracing table to keep track of the variables and their values throughout each iteration of a loop and completing the Consumer Review Lab **(Skill 2.D)**. In addition, students will learn that iteration is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units. 4.1. While Loops 4.2. For Loops 4.3. Loops and Strings 4.4. Nested For Loops 4.5. Loop Analysis
Week 2 (11) **Oct 17th to 21st**	4.6. Practice and Summary 4.7. Consumer Review Lab Methods and Control Structures 4.8 to end, AP Classroom Unit Quiz **Unit 5: Writing Classes** This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit. By the end of this unit, students will also understand the importance of documentation when writing program code. Through programming challenges and interactive activities, students will learn about commenting and conditions. Specifically, students will be able to describe pre and post conditions that are necessary for a program to work as intended **(Skill 5.D).** 5.1. Anatomy of a Java Class 5.2. Writing Constructors 5.3. Comments and Conditions 5.4. Accessor Methods 5.5. Mutator Methods
Week 3 (12) **Oct 24th to 28th** *25-27 – Book Fair* *28- Masquerade Night* *TBA-Holy Rosary Mass*	5.6. Writing Methods 5.7. Static Variables and Method 5.8. Scope and Access 5.9. this Keyword
Week 4 (13) **Oct 31st to Nov 4th** *1-All Saint's Day Mass*	5.10. Social Impacts of CS 5.11. Unit 5 Summary AP Classroom 5.14. College Board Celebrity and Data Labs (optional) Unit Quiz
Week 5 (14) **Nov 7th to 11th**	**Unit 6: Arrays** This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, students apply standard algorithms to arrays as well as identify errors in program code found in programming challenges and interactive activities throughout the unit (**Skill 4.B**). Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

	6.1. Array Creation and Access 6.2. Traversing Arrays with For Loops 6.3. Enhanced For-Loop (For-Each) for Arrays
Week 6 (15) **Nov 14th to 18th**	6.4. Array Algorithms (FRQs) 6.5 to end, AP Classroom Unit Quiz
Week 7 (16) **Nov 21st to 25th** *25 - YSC Contest* *25-Gr.12 Q2 Exam*	**Unit 7: ArrayList** 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. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised **(IOC)**. Through POGIL group work and interactive activities, students will gain an understanding of how to use computing safely and responsibly which requires being aware of privacy, security, and ethical issues. 7.1. Intro to ArrayLists 7.2. ArrayList Methods 7.3. Traversing ArrayLists with Loops 7.4. ArrayList Algorithms 7.5. Searching Algorithms 7.6. Sorting Algorithms
Week 8 (17) **Nov 28th to Dec 2nd** FYI – Pre-Exam Days *28-Gr.12 Q2 Exam*	7.7. Ethics of Data Collection and Data Privacy 7.8 to end, AP Classroom 7.12. College Board Data Lab (optional)
Week 9 (18) **Dec 5th to 9th** 8 - Foundation Day Celebrations	Unit Quiz Final Exam
Week 10 (19) **Dec 12th to 16th** **3 Days of Class** *15-16 ~Q2 Exams*	**Review / Semester 1 Project**
Dec 19th to Jan 2nd	**Christmas Break**

3rd QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 (20) **Jan 5 to 6th** **2 Days of Class**	**Unit 8: 2D Arrays** 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. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplish this is in row-major order, but it is important to cover additional traversal patterns,

	such as back and forth or column-major. After completing programming challenges and interactive exercises 8.1. Two-dimensional (2D) Arrays 8.1. Two-dimensional (2D) Arrays 8.2. Traversing 2D Arrays (nested loops) 8.3 to end, AP Classroom
Week 2 (21) **Jan 9th to 13th**	Picture lab and Steganography Lab Unit Quiz
Week 3 (22) **Jan 16th to 20th**	**Unit 9: Inheritance** Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. 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. These newly created classes are called subclasses. In this unit, students will strengthen their ability to determine an appropriate program design to solve a problem or accomplish a task **(Skill 1.A)**. Students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs. 9.1. Inheritance, Superclass, Subclass 9.2. Inheritance and Constructors 9.3. Overriding Methods 9.4. Super Keyword
Jan 23rd to 27th	**Chinese New Year**
Week 4 (23) **Jan 30th to Feb 3rd**	9.5. Inheritance Hierarchies 9.6. Polymorphism 9.7. Object Superclass 9.8 to end, AP Classroom Review and Practice Unit Quiz
Week 5 (24) **Feb 6th to 10th**	**Unit 10: Recursion** 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, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing. 10.1. Recursion 10.2. Recursive Searching and Sorting 10.3 to end, AP Classroom Unit Quiz
Week 6 (25) **Feb 13th to 17th**	Elevens Lab
Week 7 (26) **Feb 20th to 24th** *20-24 ~IOWA* *22 ~ Ash Wednesday Mass* *21-23 ~ Pre-Exam Days*	AP Review Post Test U12PE1-2 Practice Exam 1 MCQ U12PE3 Practice Exam 2 MCQ U12PE4 Practice Exam 3 MCQ’

	U12PE5 Practice Exam 4 MCQ U12PE6 Practice Exam 5 MCQ
Week 8 (27) **Feb 27th to March3rd** **3 Days of Class** 27-28 ~ 228 Memorial Day Holiday	U13 Timed practice Exams Exam 1 Review Exam 2 Review Exam 3 Review
Week 9 (28) **March 6th to 10th** **4 Days of Class** *11 – Q3 Exams*	Final Exam Review

4th QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 (29) **March 13th to 17th** **4 Days of Class** *13 – Q3 Exams* *14~ Q4 Begins*	14. Mixed Up Code - Free Response Practice 14.1. RandomStringChooser - Part A Parsons 14.1.1. Mixed Up Code Practice 14.1.2. More Mixed Up Code Practice 14.2. RandomStringChooser - Part A 14.2.1. Try and Solve It - Again 14.3. RandomStringChooser - Part B Parsons 14.4. RandomStringChooser - Part B - 2nd time 14.4.1. Try and Solve It - Again 14.5. Exercises
Week 2 (30) **March 20th to 24th** *20 ~ Fire Drill*	15. Free Response Practice 15.1. RandomStringChooser - Part A 15.1.1. Try and Solve It 15.2. RandomStringChooser - Part B 15.2.1. Try and Solve It 15.3. StringCoder - Part A 15.3.1. Try and Solve It 15.4. StringCoder - Part B 15.4.1. Try and Solve It
Week 3 (31) **March 27th to 31st**	15.5. StudentAnswerSheet - Part A 15.5.1. Try and Solve It 15.6. StudentAnswerSheet - Part B 15.6.1. Try and Solve It15.6.1. Try and Solve It 15.7. SkyView - Part A 15.7.1. Try and Solve It 15.8. SkyView - Part B 15.8.1. Try and Solve It
Apr 3rd to 14th	**Easter Break**
Week 4 (33) **Apr 17th to 21st**	15.9. Hidden Word - Write Class 15.9.1. Try and Solve It 15.10. ArrayTester - Part A 15.10.1. Try and Solve It 15.11. ArrayTester - Part B 15.11.1. Try and Solve It 15.12. NumberGroup - Part B 15.12.1. Try and Solve It 15.13. NumberGroup - Part C 15.13.1. Try and Solve It 15.14. Exercises

Week 5 (34) **Apr 24th to 28th** 24-28 ~ AP Mock Exams	Mock Exams
Week 6 (35) **May 1st to 5th** *2-4~ Pre-Exam* *1-5~ Final Exams (K, 5, 8, 12 only)* *1-5 ~ AP Exams*	AP Exams
Week 7 (36) **May 8th to 12th** *8-12~ Final Exams(K, 5, 8, 12 only)* *1-5 ~ AP Exams*	Final Exams
Week 8 (37) **May 15th to 19th** **3 Days of Class** *18-19~ Q4 Exams*	Final Exams
Week 9 (38) **May 22nd to 26th** **4 Days of Class** *22~ Record Day* *23-26 ~ Student Clearance*	Clearance
Week 10 (39) **May 29th to June 2nd** **4 Days of Class** *1 ~ Students Last Day* *2~ Teachers/Staff Meeting*	