CodeHS

Texas Computer Science 2 Course Syllabus
One Year for High School, 170 Hours

Course Overview and Goals

The Texas Computer Science 2 course is designed to foster students' creativity and innovation by presenting opportunities to design, implement, and present meaningful programs through a variety of media. By the end of this course, students will collaborate with one another, their instructor, and various electronic communities to solve problems. Through data analysis, students will identify task requirements, plan search strategies, and use computer science concepts to access, analyze, and evaluate information needed to solve problems. This course extends concepts learned in Computer Science I by deepening student understanding of system and network protocols and exposing students to the Java programming language.

Learning Environment: The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses student's mastery of the material from that unit as well as challenge problems where students can display their understanding of the material.

Programming Environment: Students write and run Java programs in the browser using the CodeHS editor.

More information: Browse the content of this course at https://codehs.com/course/20088/

Prerequisites: The Texas Computer Science 2 course is intended for students who have completed an introductory high school computer science course.

Course Breakdown

Module 1: System Administration (3-4 weeks/15-20 hours)

Students will compare and contrast common operating systems (Windows, Linux, OS) and explain the importance of application security. They will investigate security options and implement user accounts to enforce authentication and authorization. Students will also demonstrate how to work with basic and advanced command prompts.

Browse the full content of this module at https://codehs.com/course/20088/explore/module/27656

Objectives / Topics Covered	• Operating Systems • Software and Applications • Application Security • Browser Configuration • System Administration • Command Line Interface

Example Assignments	● Understanding Operating Systems ● Comparing Operating Systems ○ Installing an OS ● File Management ○ What Processor Are You Running? ● Software Licenses ● Antivirus Software ○ Data Backups ● Using Cache ● Popup Blockers ● User Accounts ○ Admin vs. Standard ● Host Security ○ Using a Log ● System Commands ○ **`cd`**, **`ls`**, **`mk`**, etc. ● Network Commands ○ **`ipconfig`**, **`netstat`**, etc.

Module 2: Networking Fundamentals (3-4 weeks/15-20 hours)

This module explores the structure and design of the internet and networks, and how this design affects the reliability of network communication, the security of data, and personal privacy. Students will learn how the Internet connects computers all over the world by use of networking protocols.

Browse the full content of this module at https://codehs.com/course/20088/explore/module/27657

Objectives / Topics Covered	● Introduction to the Internet ● Notational Systems ● Data Representation ● Internet Hardware ● Internet Addresses ● Domain Name System (DNS) ● Routing ● Packets and Protocols ● The Internet and Cybersecurity ● Creative Credit & Copyright ● Impact of the Internet
Example Assignments	● Introduction to the Internet ○ What is the Internet? What has been its impact on society? ○ Why do we need protocols for the Internet? ● Data Representation ○ Decimal to Binary ○ Hexadecimal ○ Bits to ASCII ● Internet Hardware ○ Explore how data is able to be transmitted across the ocean by using underwater cables ○ Explore the role of simple and complex networks and routers

	● Internet Addresses ○ How do IP addresses compare to postal addresses? ○ How do IP addresses work? ○ Explore the differences between IPv4 and IPv6. Why are we running out of addresses? ○ Trace a website request from the server to your computer ● Domain Name System (DNS) ○ How does DNS help with sending digital information and IP addresses? ○ Explore the process of how requesting a web resource works ● Routing ○ How is routing used to send messages/data? ○ Why is redundancy a good thing for the Internet? (fault tolerant) ● Packets and Protocols ○ Explain in your own words how a request from your computer travels through the various levels of servers to reach and return the correct webpage and resources. ○ As a class, create a protocol that will allow one classmate to send another classmate a note, without the need for talking to each other. ● The Internet and Cybersecurity ○ What are cybercrime and cyberwarfare?

Module 3: Introduction to Programming in Java with Karel the Dog (3 weeks/15 hours)
In this module, students learn the basics of Java commands, control structures, and problem-solving by solving puzzles with Karel.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27658

Objectives / Topics Covered	● Commands ● Defining vs. Calling Methods ● Designing Methods ● Program Entry Points ● Control Flow ● Looping ● Conditionals ● Classes ● Commenting Code ● Preconditions and Postconditions ● Top Down Design
Assignments / Labs	● Program-specific tasks for Karel the Dog ● **Maze Karel**: Karel is stuck in a maze. Help Karel escape and find the tennis ball at the end. Your job is to give commands to help Karel navigate the maze and end up on the tennis ball. Karel should end up facing East. ● **Teach Karel to `makePancakes()`**: Karel needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th Avenue. Each stack of pancakes should have three pancakes. Create a method called `makePancakes()` to help Karel solve this problem. ● **The Two Towers:** In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. At the end, Karel should end up on top of the second tower, facing East. ● **Super Cleanup Karel**: Karel's world is a complete mess. There are tennis

	balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing East and should clean up all of the tennis balls in the world. This program should be general enough to work in any size world with tennis balls in any location.

Module 4: Basic Java (6 weeks/30 hours)
In this module, students learn the basics of the Java programming language. This module covers printing, variables, types, and basic control structures in the Java language.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27659

Objectives / Topics Covered	- Printing - Variables - Types - Arithmetic Expressions - Casting `ints` and `doubles` - Input/Output - Errors - Loops - Conditionals - De Morgan's Laws - Short Circuit Evaluation - Debugging - Nested Control Structures - Working with the Java `String` class - Computer Ethics
Assignments / Labs	- **Add Fractions**: In this program you will ask the user for 4 integers that represent two fractions. The first number is the numerator of the first fraction. The second number is the denominator of the first fraction. The third number is the numerator of the second fraction. The fourth number is the denominator of the second fraction. Your program should add the two fractions and print out the result. - **Print the Odds**: Write a program that prints the odd numbers from 1 to 100. - **Three Strings**: Write a program that asks the user for three strings. Then, print out whether the first string concatenated to the second string is equal to the third string.

Module 5: Methods (3 weeks/15 hours)
In this module, students learn how to define methods in their programs and use auto graders to test if their methods are working correctly.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27660

Objectives / Topics Covered	- Methods - Parameters - Return values - Javadocs - `@param` - `@return` - Iterating Over Characters

	• Java Exceptions • Compile-Time vs Run-Time Exceptions • Java `String` Class and Methods • Java `Character` Class and Methods

Assignments / Labs	• **Echo**: Write a method called `echo()` that takes one String parameter called `text` and one int parameter called `numTimes` and prints out that String that number of times.
	• **Average**: Write a method called `average()` that takes two doubles and returns a double that's the average of those two numbers.
	• **Is Divisible**: Write a method that returns whether a is divisible by b. Provide proper Javadoc-style comments above the method signature.Your method signature should be
	`public boolean isDivisible(int a, int b)`
	• **First and Last**: Write a method that returns a `String` that is just the first and last character of the given string. Your return value should be only two characters long. You can assume that the given string will not be empty.The method signature should be
	`public String firstAndLast(String str)`
	• **Is it an Integer?**: Given a string, determine if it is an integer. For example, the string "123" is an integer, but the string "hello" is not. It is an integer if all of the characters in the string are digits.Return true if it is an integer, or false if it is not.
	Hint: There is a method `Character.isDigit()` that takes a char as an argument and returns a boolean value.

Module 6: Classes and Object-Oriented Programming (4 weeks/20 hours)

This module teaches students the basics of Object Oriented Programming in Java, which is a powerful programming paradigm. Students will learn how objects store data and interact with each other in Java programs. Students will design and implement classes and extend classes using inheritance.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27661

Objectives / Topics Covered	• Using Classes as a Client • Classes vs Objects • Class Methods • Instance Variables • Constructors • Visibility • Information Hiding • `this` • `static` • `super` • The Java `Math` Class and Methods • Creating Random Values • Designing Classes • Creating Classes • Getter and Setter Methods • Inheritance • Method Overloading

	• Local Variables and Scope • Comparing Objects vs Primitive Types • Abstract Classes • Packages • Polymorphism • Interfaces • Modifying Classes • `Object` is the Superclass
Assignments / Labs	• **Fraction Class**: Design and implement a `Fraction` class from scratch, including a constructor, getter and setter methods, a `toString` method, and methods to add, subtract, and multiply by other `Fraction` objects. • **Rock, Paper, Scissors**: Implement a `RockPaperScissors` class with a `getWinner(String user, String computer)` method that allows a user to play the game Rock, Paper, Scissors against a computer that picks moves randomly. • **Abstract Methods**: Add an abstract method to an existing `Shape` class called `public abstract double getPerimeter()` and implement this method on each of the Shape subclasses, `Square`, `Rectangle`, `Pentagon`, and `Circle` • **Fun with Solids**: Given the `Solid` abstract class, extend it with: `Pyramid`, `Cylinder`, `RectangularPrism`, and `Sphere`. Make sure to create the constructor, `volume(),` and `surfaceArea()` methods for each class.

Module 7: Data Structures (4 weeks/20 hours)
In this module, students learn basic data structures in Java including arrays, ArrayLists, 2-dimensional arrays, and HashMaps. Data structures will be used to design larger applications.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27662

Objectives / Topics Covered	• Declaring and Initializing Arrays • Constructing `ArrayLists` • Indexing Into `Arrays/ArrayLists` • Iterating Over `Arrays/ArrayLists` • Getting the Length of an `Array/ArrayLists` • `ArrayIndexOutOfBoundsException` • `IndexOutOfBoundsException` • Object References • `Arrays/ArrayLists` as Parameters • `Arrays/ArrayLists` as Return Values • Inserting and Deleting Elements • Wrapper Classes • Storing Objects in `Arrays` vs. `ArrayLists` • Numerical Representations of Integers • The `List` Interface • Declaring and Initializing 2D Rectangular Arrays • Using Nested Loops to Iterate through 2D Arrays • Row-Major order • `HashMaps`

Assignments / Labs	● **First Element:** Write a method that returns the first value in an `ArrayList`
	● **Under the Hood**: See how an `ArrayList` works under the hood. Write an `ExpandingArray` class that stores an array as an instance variable and supports the methods
	`public void add(int index, int element)`
	`public void add(int element)`
	`public int remove(int index)`
	`public int size()`
	`public String toString()`
	● **Sum 2D Array**: Write the method `public int sumRow(int[][] matrix, int row)`, which sums the rows in the 2D array called matrix.
	● **Black Jack**: Explore and add to the code for a `BlackJack` game with a `Card` class, `Deck` class, `Hand` class, and `BlackJack` class
	● **Battleship**: Implement the game Battleship with several incremental checkpoints
	○ Implement the `Ship` class
	○ Implement the `Location` class
	○ Implement the `Grid` class
	○ Implement adding a `Ship` to a `Grid`
	○ Design and implement the `Player` class
	○ Design and implement the `Battleship` class
	○ Add extra features to the game

Module 8: Steganography Lab (1 week/5 hours)

Steganography is the practice of concealing messages or information within other non-secret text or data. Students will use the same code from Picture Lab to explore the concepts of steganography and 2D arrays, hiding images or text inside of other images.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27663

Objectives / Topics Covered	● Exploring Colors ● Clearing Bits ● Changing Colors ● Setting Bits ● Bits vs Vectors ● Hiding and Revealing Hidden Pictures ● Hiding and Revealing Hidden Messages
Assignments / Labs	● **Setting and Clearing Bits**: Students learn how to encode information within pixels of an image. ● **Hiding and Revealing a Picture**: Students hide pictures within pictures by changing the value of pixels within a picture. ● **Identifying a Hidden Picture**: Students identify if a picture has a hidden message encrypted within it. ● **Hiding and Revealing a Hidden Message**: Students learn to encrypt and decrypt messages within a picture.

Module 9: Algorithms and Recursion (3 weeks/15 hours)

In this module, students will be introduced to fundamental searching and sorting algorithms including sequential search, binary search, insertion sort, selection sort, and mergesort, as well as the important concept of recursion.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27664

Objectives / Topics Covered	● Implementing and using Sequential Search ● Implementing and using Binary Search ● Comparing relative run times of Sequential and Binary Search ● Brief introduction to Big-Oh ● Counting comparisons in searches and sorts ● Insertion Sort ● Selection Sort ● Merge Sort ● Pros and cons of each sorting algorithm ● Divide and Conquer ● Recursion ● `java.util.Arrays` ● Sorting and searching with both arrays and `ArrayLists`
Assignments / Labs	● Interactive visualization to explore how each sorting algorithm works ● Several exercises to master the topics above. 8 in total ● Example Exercises ○ Implement a sequential search function that takes an `ArrayList<Integer>` as a parameter ○ Modify the sequential search and binary search functions to return the number of comparisons made in each search, which one is more efficient? Test your functions on arrays of various sizes. ○ Modify insertion sort to sort elements in descending order ○ Write a recursive function to reverse a string ○ Mergesort is a complicated algorithm, but how complicated is it? In this exercise, we'll be taking our example code from before and adding a cool feature: at every recursive step, print out to the console what the two halves are that are going to be merged together.

Module 10: Celebrity Lab (1 week/5 hours)

Students will discuss class design as it relates to the game Celebrity, where a person or team tries to guess the name of a celebrity from a given clue or set of clues. This lab includes inheritance as the basis for one of the activities and also includes a Graphical User Interface.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27665

Objectives / Topics Covered	● Creating GUIs ● JFrame and JSwing
Assignments / Labs	● Putting it All Together ○ Students learn how to create a GUI with interactive buttons. Students create GUI components that are used in a game. ● Extending the Celebrity Class ○ Students extend existing classes to make their game more customizable.

Module 11: Final Project (2-4 weeks/10-20 hours)

Students will be tasked with creating a website of their own choosing. The website will have to follow specific criteria - a certain number of pages, responsiveness, and use of APIs. Students will go through a feedback process, and learn about making their websites more accessible to a wide array of users.

Browse the full content of this unit at https://codehs.com/course/20088/explore/module/27666

Objectives / Topics Covered	● Planning Your Innovation ● Providing Feedback
Assignments / Labs	● Version Control ○ Students create a running version document that tracks the changes they make to their innovation. Students learn how to catalog each version, and are asked to document how the website changes over time. ● Present your Innovation ○ Students are required to make a presentation highlighting the innovation that they created, and how it addresses a particular problem in their community. Students highlight how their innovation changed over the course of development and as a result of feedback provided by user testing.

Module 12: Computer Science Careers (1 week/5-7 hours)

Students learn about a variety of computer science careers and organizations, and what the next steps could look like for them if interested.

Objectives / Topics Covered	● Careers and internships ● CS career preparation ● Legal and ethical responsibilities ● Workplace readiness
Example Assignments / Labs	● Exploring computer science careers, internships, and organizations ● Learning about CS resumes and certifications ● Researching about a major ethical or legal topic in CS ● Reflecting on what it means to be a leader and the skills required to be successful in the workplace