

Module 2: Networking Fundamentals (3-4 weeks)

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>

<div>Objectives / Topics Covered</div> <div>8/12/24 – 09/06/24</div>	<ul style="list-style-type: none">● 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
<div>Example Assignments</div>	<ul style="list-style-type: none">● Introduction to the Internet<ul style="list-style-type: none">○ What is the Internet? What has been its impact on society?○ Why do we need protocols for the Internet?● Data Representation<ul style="list-style-type: none">○ Decimal to Binary○ Hexadecimal○ Bits to ASCII● Internet Hardware<ul style="list-style-type: none">○ 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

	<ul style="list-style-type: none"> ● 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 (7 weeks)

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>

<p>Objectives / Topics Covered</p> <p>09/09/24 – 10/25/24</p>	<ul style="list-style-type: none"> ● ● 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	<ul style="list-style-type: none"> ● 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 10/28/24 – 11/15/24	<ul style="list-style-type: none"> ● ● Methods Parameters ● Return values ● Javadocs ● @param ● @return ● Iterating Over Characters
	<ul style="list-style-type: none"> ● Java Exceptions ● Compile-Time vs Run-Time Exceptions ● Java String Class and Methods ● Java Character Class and Methods
Assignments / Labs	<ul style="list-style-type: none"> ● Echo: Write a method called <code>echo()</code> that takes one String parameter called <code>text</code> and one int parameter called <code>numTimes</code> and prints out that String that number of times. ● Average: Write a method called <code>average()</code> that takes two doubles and returns a double that's the average of those two numbers. ● Is Divisible: Write a method that returns whether <code>a</code> is divisible by <code>b</code>. Provide proper Javadoc-style comments above the method signature. Your method signature should be <code>public boolean isDivisible(int a, int b)</code> ● 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 <code>public String firstAndLast(String str)</code> ● 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 <code>Character.isDigit()</code> that takes a char as an argument and returns a boolean value.

Module 6: Classes and Object-Oriented Programming (6 weeks)

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>

<p>Objectives / Topics Covered</p> <p>11/18/24 -01/24/25</p>	<ul style="list-style-type: none">● 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
	<ul style="list-style-type: none">● Local Variables and Scope● Comparing Objects vs Primitive Types● Abstract Classes● Packages● Polymorphism● Interfaces● Modifying Classes● Object is the Superclass

Assignments / Labs	<ul style="list-style-type: none"> • Fraction Class: Design and implement a <code>Fraction</code> class from scratch, including a constructor, getter and setter methods, a <code>toString</code> method, and methods to add, subtract, and multiply by other <code>Fraction</code> objects. • Rock, Paper, Scissors: Implement a <code>RockPaperScissors</code> class with a <code>getWinner(String user, String computer)</code> 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 <code>Shape</code> class called <code>public abstract double getPerimeter()</code> and implement this method on each of the <code>Shape</code> subclasses, <code>Square</code>, <code>Rectangle</code>, <code>Pentagon</code>, and <code>Circle</code> • Fun with Solids: Given the <code>Solid</code> abstract class, extend it with: <code>Pyramid</code>, <code>Cylinder</code>, <code>RectangularPrism</code>, and <code>Sphere</code>. Make sure to create the constructor, <code>volume()</code>, and <code>surfaceArea()</code> methods for each class.
--------------------	--

Module 7: Data Structures (4 weeks)

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 01/27/25 – 02/28/25	<ul style="list-style-type: none"> • Declaring and Initializing Arrays Constructing • <code>ArrayLists</code> • Indexing Into Arrays/<code>ArrayLists</code> • Iterating Over Arrays/<code>ArrayLists</code> • Getting the Length of an Array/<code>ArrayLists</code> • <code>ArrayIndexOutOfBoundsException</code> • <code>IndexOutOfBoundsException</code> • Object References • Arrays/<code>ArrayLists</code> as Parameters • Arrays/<code>ArrayLists</code> as Return Values • Inserting and Deleting Elements • Wrapper Classes • Storing Objects in Arrays vs. <code>ArrayLists</code> • Numerical Representations of Integers • The <code>List</code> Interface • Declaring and Initializing 2D Rectangular Arrays • Using Nested Loops to Iterate through 2D Arrays
--	---

	<ul style="list-style-type: none"> • Row-Major order • HashMaps
Assignments / Labs	<ul style="list-style-type: none"> • 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 <code>public void add(int index, int element)</code> <code>public void add(int element)</code> <code>public int remove(int index)</code> <code>public int size()</code> <code>public String toString()</code> • Sum 2D Array: Write the method <code>public int sumRow(int[][] matrix, int row)</code>, 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 <ul style="list-style-type: none"> ○ 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 03/03/25 – 03/07/25	<ul style="list-style-type: none"> • Exploring Colors Clearing • Bits • Changing Colors • Setting Bits • Bits vs Vectors • Hiding and Revealing Hidden Pictures • Hiding and Revealing Hidden Messages
--	---

Assignments / Labs	<ul style="list-style-type: none"> ● 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 (4 weeks)

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>

<p>Objectives / Topics Covered</p> <p>03/17/25 – 04/11/25</p>	<ul style="list-style-type: none"> ● 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 ● <code>java.util.Arrays</code> ● Sorting and searching with both arrays and <code>ArrayLists</code>
Assignments / Labs	<ul style="list-style-type: none"> ● Interactive visualization to explore how each sorting algorithm works ● Several exercises to master the topics above. 8 in total

	<ul style="list-style-type: none"> ● Example Exercises <ul style="list-style-type: none"> ○ Implement a sequential search function that takes an <code>ArrayList<Integer></code> 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	<ul style="list-style-type: none"> ● Creating GUIs ● JFrame and JSwing
Assignments / Labs 4/14/25 – 04/18/25	<ul style="list-style-type: none"> ● Putting it All Together <ul style="list-style-type: none"> ○ Students learn how to create a GUI with interactive buttons. Students create GUI components that are used in a game. ● Extending the Celebrity Class <ul style="list-style-type: none"> ○ Students extend existing classes to make their game more customizable.

Module 11: Final Project (2-4 weeks)

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	<ul style="list-style-type: none"> ● Planning Your Innovation Providing ● Feedback
Assignments / Labs 04/21/25 – 05/16/25	<ul style="list-style-type: none"> ● Version Control <ul style="list-style-type: none"> ○ 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 <ul style="list-style-type: none"> ○ 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 05/19/25 – 05/23/25	<ul style="list-style-type: none">● Careers and internships CS● career preparation● Legal and ethical responsibilities● Workplace readiness
Example Assignments / Labs	<ul style="list-style-type: none">● 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