



# CodeHS

## Creative Computing Course Syllabus

**1 semester for MS or HS (80-95 contact hours)**

### Course Overview and Goals

The Creative Computing course is a first year computer science course introducing the basics of programming with Karel the Dog and the basics of designing a web page. Students will learn to code using blocks to drag and drop, but they can switch between blocks and text as desired. Students will create a personal portfolio website showing projects they build throughout the course.

With a unique focus on creativity, problem solving and project based learning, Creative Computing gives students the opportunity to explore several important topics of computing using their own ideas and creativity to develop an interest in computer science that will foster further endeavors in the field.

**Learning Environment:** The course utilizes a blended classroom approach. The content is a mix of web-based and physical activities. Students will write and run code in the browser, create websites and digital presentations, and engage in in-person collaborative exercises with classmates. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students.

**Programming Environment:** Students write and run programs in the browser using the CodeHS online editor. Students will be able to write both text based and block based programs in Karel. Students will also create several webpages using HTML and CSS. These webpages will be hosted on the CodeHS website so that they can keep a running portfolio of their creative projects, and easily share their programs with the world. Students gain programming experience early on in the course that will enable them to explore the rest of the course topics through computational thinking practices.

**Quizzes:** Each lesson includes at least one formative short multiple choice quiz. At the end of each unit, students take a summative multiple choice unit quiz that assesses their knowledge of the concepts covered in the unit.

**Prerequisites:** The Creative Computing course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for those new to computer science.

**More information:** Browse the content of this course at <https://codehs.com/course/2349>

## Course Breakdown

### Unit 1: Web Design (6 weeks/30 hours)

Browse the full content of this unit at <https://codehs.com/library/course/2349/module/3456>

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> <li>● Creating webpages using HTML             <ul style="list-style-type: none"> <li>○ Links</li> <li>○ Images</li> <li>○ Lists</li> <li>○ Tables</li> <li>○ Inline styling</li> </ul> </li> <li>● Styling webpages with CSS             <ul style="list-style-type: none"> <li>○ Creating CSS rules</li> <li>○ CSS classes</li> <li>○ CSS IDs</li> </ul> </li> <li>● How webpages are requested and delivered</li> </ul>
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> <li>● Links             <ul style="list-style-type: none"> <li>○ Learn how to link different pages together</li> <li>○ Example Exercise: Create a webpage that provides links to your favorite books</li> </ul> </li> <li>● Images             <ul style="list-style-type: none"> <li>○ Learn how to add and format images</li> <li>○ Example Exercise: Create a collage of images</li> </ul> </li> <li>● Tables             <ul style="list-style-type: none"> <li>○ Learn how to create and style tables</li> <li>○ Example Exercise: Create a table describing your favorite music artists and songs</li> </ul> </li> <li>● Styling with CSS             <ul style="list-style-type: none"> <li>○ Use CSS to add background colors, font colors, font styles, borders, and position elements on the page</li> <li>○ Example Exercise: Create CSS classes and IDs to apply formatting to a BINGO board</li> <li>○ Example Exercise: Create CSS classes to style a music library web page</li> <li>○ Example Exercise: Create CSS Rules to put a Karel puzzle together</li> </ul> </li> <li>● Viewing websites             <ul style="list-style-type: none"> <li>○ How are web pages served to your computer?</li> <li>○ Where do web pages live?</li> </ul> </li> <li>● Final Project             <ul style="list-style-type: none"> <li>○ Build your own homepage using everything you've learned in the module</li> <li>○ This homepage will serve as your personal portfolio of creative projects as you continue through the course!</li> </ul> </li> </ul>

## Unit 2: Introduction to Programming with Karel the Dog (10 weeks/50 hours)

Browse the full content of this unit at <https://codehs.com/library/course/2349/module/3467>

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> <li>● Commands</li> <li>● Defining versus Calling Methods</li> <li>● Designing methods</li> <li>● Control flow</li> <li>● Looping</li> <li>● Conditionals</li> <li>● Commenting code</li> <li>● Preconditions and Postconditions</li> <li>● Top Down Design</li> </ul>
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> <li>● Commands <ul style="list-style-type: none"> <li>○ Program-specific tasks for Karel the Dog</li> <li>○ Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.</li> </ul> </li> <li>● Functions <ul style="list-style-type: none"> <li>○ Teach Karel new commands like <code>turnRight()</code> or <code>makePancakes()</code></li> <li>○ Example Exercise: Pancakes Karel is the waiter. He 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 <code>makePancakes()</code> to help Karel solve this problem.</li> </ul> </li> <li>● Top Down Design <ul style="list-style-type: none"> <li>○ Solve large Karel problems by breaking them down into smaller, more manageable problems</li> <li>○ Example Exercise: 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.</li> </ul> </li> <li>● Loops and Conditionals <ul style="list-style-type: none"> <li>○ Example Exercise: Random Hurdles Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.</li> <li>○ Example Exercise: 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.</li> </ul> </li> </ul>

	<p>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 on any size world with tennis balls in any locations.</p>
--	---

### Unit 3: Introduction to Programming in Python with Tracy the Turtle (3 weeks/15 hours)

Browse the full content of this unit at <https://codehs.com/library/course/2349/module/3468>

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> <li>● Commands</li> <li>● Colors</li> <li>● Loops</li> <li>● Functions</li> </ul>
<p>Assignments / Labs</p>	<ul style="list-style-type: none"> <li>● Commands <ul style="list-style-type: none"> <li>○ Program-specific tasks for Tracy the Turtle</li> <li>○ Example Exercise: Staircase Write a program to have Tracy draw a staircase. Use the built-in commands <code>forward</code>, <code>left</code> and <code>right</code>.</li> </ul> </li> <li>● Functions <ul style="list-style-type: none"> <li>○ Teach Tracy new commands to simplify programs and avoid repeated code.</li> <li>○ Example Exercise: Fenceposts Tracy has been hired to build a fence around someone's yard! Make a function called <code>make_fencepost</code> that draws a fencepost of height 30. The fencepost should have lines of length 10 on either side. Then, use a for loop and your new <code>make_fencepost</code> function to make a fence with five fence posts.</li> </ul> </li> <li>● Loops <ul style="list-style-type: none"> <li>○ Example Exercise: Circle Have Tracy draw a circle. You can do this by making a polygon that has 360 sides, each of length 1. This ends up looking more or less like a circle.</li> </ul> </li> </ul>