



CodeHS

South Carolina Fundamentals of Computer Science Course Syllabus

Course Overview and Goals

This course is designed to introduce students to the field of computer science through an exploration of engaging and accessible topics. Through creativity and innovation, students will use critical thinking and problem solving skills to implement projects that are relevant to students' lives. They will create a variety of computing artifacts while collaborating in teams. Students will gain a fundamental understanding of the history and operation of computers, programming, and web design. Students will also be introduced to computing careers and will examine societal and ethical issues of computing.

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.

Programming Environment: For the programming units, students write and run JavaScript programs in the browser using the CodeHS editor. For the web design unit, students write HTML and CSS code in the browser using the CodeHS online editor. Classes can choose to write code using either blocks or text.

More information: Browse the content of this course at <https://codehs.com/course/6379>

Prerequisites

The Fundamentals of Computer Science 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 new coders.

Course Breakdown

The full standards alignment can be found at https://codehs.com/standards/framework/SC_FCS/course/6379

Unit 1: Digital Citizenship and Cyber Hygiene (5 weeks/25 hours)

This unit focuses on why cybersecurity is important, recent threats to cybersecurity, and different careers in the field. It includes topics on Internet etiquette and how to stay safe on the world wide web. Students will also look at the potential effects of their digital footprints, how to protect information from online risks, and the implications of cyberbullying. Finally, the module includes how to find and cite quality resources online.

Main Standards:

- C.2 Identify proper netiquette when using e-mail, social media, and other technologies for communication purposes.
- C.3 Identify potential abuse and unethical uses of laptops, tablets, computers, and/or networks.
- C.4 Explain the consequences of social, illegal, and unethical uses of technology (e.g., cyberbullying, piracy; illegal downloading; licensing infringement; inappropriate uses of software, hardware, and mobile devices in the work environment).
- C.5 Discuss legal issues and the terms of use related to copyright laws, fair use laws, and ethics pertaining to downloading of images, photographs, documents, video, sounds, music, trademarks, and other elements for personal use.
- C.6 Describe ethical and legal practices of safeguarding the confidentiality of business-and personal-related information.
- C.7 Describe possible threats to a laptop, tablet, computer, and/or network and methods of avoiding attacks.
- K.1 Examine the consequences resulting from issues involving ethics around security, privacy, copyright, fair use, intellectual property, social media and licensing.
- K.2 Explain the importance of Acceptable Use Policies.
- K.3 Explain the importance of data security and physical security.
- K.4 Explain the concepts of confidentiality, integrity, and availability (CIA).
- K.5 Identify computing threats (e.g., spyware, adware, malware, viruses, ransomware, phishing, hacking, software piracy, identity theft, etc.) and their potential impacts on society.
- K.6 Explain the concept of encryption and how it is used on a daily basis.

Browse the full content of this module at <https://codehs.com/library/course/6379/module/9148>

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> ● What is Cybersecurity? ● The CIA Triad ● Digital Footprint and Reputation ● Cyberbullying ● Internet Safety ● Privacy and Security ● Information Literacy ● Visualizing and Interpreting Data ● Data Collection and Limitations ● Creative Credit and Copyright ● Hacking Ethics ● Project: PSA
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> ● What is Cybersecurity? <ul style="list-style-type: none"> ○ Cybersecurity defined ○ Why is cybersecurity important? ○ Cybersecurity in the news ○ Cybersecurity and IoT (Internet of Things) ○ How do we prevent cyber attacks? ○ Example activities: <ul style="list-style-type: none"> ■ Summarize and discuss recent cyber attacks ■ Explore a threat map to see where cyber attacks are coming from and which countries are being targeted ● The CIA Triad

- What is the CIA triad? (confidentiality, integrity, availability)
- What are “secure systems?”
- What do confidentiality, integrity, and availability mean in cybersecurity?
- Example activities:
 - Determine where scenarios break part of the CIA Triad
- Digital Footprint and Reputation
 - What is a digital footprint?
 - What is **your** digital footprint and reputation?
 - What does it mean that the internet is public and permanent?
 - Who looks at your digital footprint and reputation?
 - What are some recommended social media guidelines?
 - How can you maintain your digital footprint?
 - What does your digital footprint say about you?
 - Example activities:
 - What is your digital footprint?
 - Are you going to make any changes in what you post on social media?
- Cyberbullying
 - What is cyberbullying?
 - What are the impacts of cyberbullying?
 - Are there cyberbullying roles?
 - What do you do if you are being bullied?
 - What do you do if you see bullying?
 - How can you be an upstander?
 - Example activities:
 - Explore cyberbullying scenarios: What would you do?
- Internet Safety
 - What are some ways to stay safe online?
 - What are some online safety guidelines?
 - Example activities:
 - Explore Internet safety scenarios: What would you do?
- Privacy and Security
 - What are data privacy and security?
 - How can you keep personal data secure and private?
 - What can happen if your data is stolen and what can you do about it?
 - Example activities:
 - Test out various passwords on a site
 - Explore Google’s privacy policy: What do they know about you?
- Information Literacy
 - What is information literacy?
 - How can you do effective internet searches?
 - What are some techniques for judging source legitimacy and identifying misinformation?
 - Example activities:

	<ul style="list-style-type: none"> <ul style="list-style-type: none"> ■ Create and test search queries ■ Explore evidence for using sources ● Creative Credit and Copyright <ul style="list-style-type: none"> ○ What is copyright? ○ What are the different types of copyright licenses ○ Example activities: <ul style="list-style-type: none"> ■ Create citations for sources ■ Explore image search tools ● Hacking Ethics <ul style="list-style-type: none"> ○ What are hackers? ○ Are there different kinds of hackers? (white, black, grey) ○ What are bug bounty programs? ○ Is hacking always illegal? ○ What are the consequences of illegal hacking? ○ Example activities: <ul style="list-style-type: none"> ■ Explore what penetration testing is ■ Sign an ethical hacker agreement ● Final project: Create a Public Service Announcement <ul style="list-style-type: none"> ○ Create a Public Service Announcement (PSA) to teach your peers about your selected topic in digital citizenship and cyber hygiene. You can select any of the topics covered in this module. Be creative and make it fun! You could make a video, song, poster, or slideshow.
--	--

Unit 2: What is Computing? (2 weeks/10 hours)

This unit addresses the question What is Computing? Students review a history of computing, learn about the various parts that make up modern computers, learn about the impact computing has had on today's world, and learn about the impacts computing could potentially have in the future.

Main Standards:

- C.8 Evaluate various solutions to common hardware and software problems.
- F.3 Discuss the progression of computing and explain its impact on society (e.g. hardware, programming languages, applications, Internet, emerging technologies, etc.).
- F.4 Explain Moore’s Law.
- G.1 Identify and define the key functional components (input devices, output devices, processor, operating system, software applications, memory, storage, etc.).
- G.2 Understand the terms and units that are used to describe major hardware components (RAM, ROM, Peta-, Tera-, Giga-, Mega- Kilo-, Hz, Bit, Byte, Binary, etc.).
- G.3 Describe how software and hardware interact.
- G.6 Compare and contrast operating systems (e.g., Mac, Windows, Linux, ChromeOS, iOS, Android).
- G.7 Evaluate hardware and software configuration to meet situational and budgetary requirements (e.g. gaming, Internet browsing, student, graphic designer, etc.).
- G.8 Make hardware and software recommendations to improve a computer system.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9228>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● History of Computers ● Computer Organization
-----------------------------	---

	<ul style="list-style-type: none"> ● Software ● Hardware ● Future of Computing ● Final Project
<p>Example Assignments / Labs</p>	<ul style="list-style-type: none"> ● History of Computers <ul style="list-style-type: none"> ○ Find out when the first computers were created ○ Research famous computer innovators ○ What roles do computers play in your life? ○ Example Activity: <ul style="list-style-type: none"> ■ Summarize an era of advances in computers ● Computer Organization <ul style="list-style-type: none"> ○ What parts do modern computers have? ○ What are input devices? ○ What are output devices? ○ Example Activity: <ul style="list-style-type: none"> ■ Draw a computer and label all of its parts, including the input devices and output devices ● Software/Hardware <ul style="list-style-type: none"> ○ What's the difference? ○ What hardware components make up a computer? ○ What is software used for? ○ Example Activity: <ul style="list-style-type: none"> ■ Label the parts of your computer ● Future of Computing <ul style="list-style-type: none"> ○ Research uses of Artificial Intelligence in use now ○ Research new ways of storing data ○ Example Class Activity: <ul style="list-style-type: none"> ■ In what ways can we use technology that we couldn't 10 years ago. Are these technological advances helpful or harmful overall? ● Final Project <ul style="list-style-type: none"> ○ Create a presentation about a computer. Choose any computer -- a phone, an early computer model, drones, etc. Who built it and why? How does it interact with people? How do people interact with the computer?

Unit 3: Programming with Karel (7 weeks, 35 hours)

Karel is a dog that only knows how to move, turn left, and place tennis balls in his world. Students can give Karel these commands to instruct him to do certain things. Karel is used to show students what it means to program, and allows them to focus on problem solving.

Main Standards:

- H.2 Solve a problem by applying appropriate problem solving techniques (understand the problem, plan the solution, carry out the plan, review and discuss)
- H.3 Define algorithm (a set of clearly defined, logical steps to solve a problem).
- H.4 Demonstrate an understanding of algorithms and their practical applications.
- H.5 Create, evaluate, and adjust algorithms to solve a variety of problems.
- I.1 Express the design of a program using representations such as flowcharts and pseudocode.
- I.2 Analyze and explain how a particular program functions.

- I.3 Solve problems of increasing complexity using visual block-based programming individually and collaboratively.
- I.4 Write code that uses variables, events, functions, operators (i.e. arithmetic, relational, logical), conditional control structures (e.g., if, if-else) and repetition/iteration control structures (e.g., while, for).
- I.6 Edit, compile/run, test, and debug a program.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9149>

<p>Objectives / Topics Covered</p>	<ul style="list-style-type: none"> ● Commands ● Defining vs. Calling Methods ● Designing methods ● Program entry points ● Control flow ● Looping ● Conditionals ● Classes ● Commenting code ● Preconditions and Postconditions ● Top Down Design
<p>Assignments / Labs</p>	<ul style="list-style-type: none"> ● 30 Karel Programming Exercises and Challenges in total ● Program-specific tasks for Karel the Dog <ul style="list-style-type: none"> ○ 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. ● Teach Karel new commands like <code>turnRight()</code> or <code>makePancakes()</code> <ul style="list-style-type: none"> ○ 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. ● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design <ul style="list-style-type: none"> ○ 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. ● Using control structures and conditionals to solve general problems <ul style="list-style-type: none"> ○ 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.

Unit 4: Karel Challenges (2 weeks, 10 hours)

In this unit, students take all the foundational concepts from Karel to solve some programming challenges.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9150>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Solving large and more complex problems using Karel
Assignments / Labs	<ul style="list-style-type: none"> ● Several Karel challenges to tie everything learned in the Karel module together <ul style="list-style-type: none"> ○ 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. 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.

Unit 5: Networks and the Internet (3 weeks/15 hours)

This unit explores the structure and design of the internet, and how this design affects the reliability of network communication, the security of data, and personal privacy.

Main Standards:

- F.1 Define key computing terms (e.g. hardware, software, data, etc.).
- G.2 Understand the terms and units that are used to describe major hardware components (RAM, ROM, Peta-, Tera-, Giga-, Mega- Kilo-, Hz, Bit, Byte, Binary, etc.).
- G.4 Discuss how and why binary is used to represent data in a computer.
- G.5 Describe how a picture, sound/song, and video are digitized and represented in a computer.
- H.1 Describe how computer programs and apps can be used to solve various problems (e.g., desktop, mobile, enterprise).
- J.2 Describe how Web sites are used to communicate and exchange data.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9151>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Structure of the internet ● How network data is transmitted ● Hardware involved in the transmission of data ● How the internet has impacted everyday life
Example Assignments / Labs	<ul style="list-style-type: none"> ● Structure of the internet <ul style="list-style-type: none"> ○ Explore the differences between IPv4 and IPv6. Why are we running out of addresses? ○ Explore the different levels of the internet. ○ Example Activity <ul style="list-style-type: none"> ■ Trace a website request from the server, through the network, and to your computer ● How data is transmitted <ul style="list-style-type: none"> ○ How are internet packets able to find their way to your computer? ○ 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? ○ Example Activity: <ul style="list-style-type: none"> ■ As a class, create a protocol that will allow one classmate to send another classmate a note,

	<p>without the need for talking to each other.</p> <ul style="list-style-type: none"> ● Hardware involved <ul style="list-style-type: none"> ○ Explore the role of routers ○ Why are protocols so important? ○ Explore how data is able to be transmitted across the ocean by using underwater cables
--	--

Unit 6: Javascript & Graphics (2 weeks, 10 hours)

This unit Introduces students to the basics of JavaScript, including variables, user input, mathematics, basic graphics, and image representations.

Main Standards:

- I.5 Differentiate between text and numerical data.
- E.3 Demonstrate mathematical reasoning.
- E.4 Demonstrate job-specific mathematics skills.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9152>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Variables ● User Input ● Arithmetic Expressions ● Graphics ● Image Representation
Assignments / Labs	<ul style="list-style-type: none"> ● Using variables and getting user input using JavaScript <ul style="list-style-type: none"> ○ Example Exercise: Dinner Plans Prompt the user for their name, then ask them what time you should meet for dinner. Greet them by name and tell them you will meet them at the time they specified! ● Create basic graphics in JavaScript <ul style="list-style-type: none"> ○ Example Exercise: Flag of the Netherlands This program should draw the flag for the Netherlands. The top third of the canvas is red, the middle third is white, and the bottom third is blue, as shown below. You will need to use Rectangle objects in this program.

Unit 7: Graphics Challenges - (2 weeks, 10 hours)

In this unit, students learn how to pair-program! They will also take all the foundational concepts from JavaScript Graphics to solve some programming challenges.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9153>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Pair programming ● Solving large and more complex problems that incorporate graphics and user input
Assignments / Labs	<ul style="list-style-type: none"> ● Several graphics challenges to tie everything learned in the module together <ul style="list-style-type: none"> ○ Example Exercise: Ghost

	Write a program to draw a ghost on the screen. You must do this by using the constant values given (this will allow us to easily alter the size or color of the ghost.)
--	---

Unit 8: Web Development (3 weeks, 15 hours)

In this unit, students will go through a high level introduction to HTML, CSS, and the processes involved in viewing web pages on the internet. Students will create several simple web pages using the CodeHS online editor to gain practice using the various features of HTML and CSS. This unit culminates with each student making their own website about themselves, hosted on their own custom CodeHS URL.

Main Standards:

- J.3 Plan a web page considering subject, devices, audience, layout, color, links and graphics.
- J.4 Create a web page that contains a variety of HTML elements (e.g., hyperlinks, ordered and unordered lists, images, headings, paragraph) and CSS styles.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9373>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● HTML ● Formatting Text ● Links ● Images ● Lists & Tables ● CSS ● CSS by Tag, Class, and ID
Assignments / Labs	<ul style="list-style-type: none"> ● What is HTML? Why is it important? ● HTML tags ● Formatting ● Lists ● Tables ● CSS Selectors ● Separating style from structure ● Using CSS to style your own websites ● Make your own Homepage! <ul style="list-style-type: none"> ○ Students will build their own website about themselves.

Unit 9: Computer Science Careers (1 week, 5 hours)

In this short unit, students take some time to explore and discover different computer science careers. They will also examine inclusive coding and how to avoid bias in computer programming.

Main Standards:

- F.2 Identify key individuals and their impact on the field of computing.
- L.1 Compare and contrast the five disciplines of computing: computer science, software engineering, information technology, information systems, and computer engineering.
- L.2 Compare and contrast careers in computing along with their education, training requirements, industry certifications and salary ranges.
- L.3 Identify gender, diversity and geographic related issues in computing.
- L.4 Describe how computing enhances other career fields.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9158>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Computer Science Careers ● Computer Science used in non CS Careers ● Inclusive Coding
Assignments / Labs	<ul style="list-style-type: none"> ● Career Research ● Computer Science concepts that are used in non computer science positions ● How can a computer program's bias become dangerous? ● Why is it important to have a more diverse group of people in the computer science field?

Unit 10: Final Project (2-4 weeks, 10-20 hours)

This unit introduces students to the theory and practice of user interface design. Students learn about what makes an engaging and accessible user interface, and will employ an iterative design process including rapid prototyping and user testing to design and develop their own engaging program.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9157>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Project Planning ● Final Project Creation
Assignments / Labs	<ul style="list-style-type: none"> ● Program Requirements: <ul style="list-style-type: none"> ○ must use JavaScript Graphics to tell a story or give information about an important topic ○ Look back at the PSA project you completed earlier in the course if you need some inspiration! ○ must allow the user to interact with your project with their mouse ○ must use at least one timer ○ must break down the program into multiple functions ○ must utilize control structures where applicable ● Alternative Storytelling <ul style="list-style-type: none"> ○ Linear story <ul style="list-style-type: none"> ■ use a timer to switch scenes at a specific interval ■ utilize a button to switch scenes when the user decides to move on ● Choose-your-own-Adventure <ul style="list-style-type: none"> ○ the user is given a selection and the story plays out based on the option they choose ● Moving Scenes <ul style="list-style-type: none"> ○ use a timer to have different parts of your scene move around the screen to add to the user experience

Supplementary Unit Guide:

These units can be used during the course for added practice or after the course has been completed for further review.

Supplementary Unit	Prerequisite/Recommended Unit(s)	# of activities
<i>Computer Science Careers</i>	No prerequisites	8
<i>Extra Karel Practice</i>	Programming with Karel Karel Challenges	12
<i>Extra Karel Puzzles</i>	Programming with Karel Karel Challenges	11
<i>Functions and Parameters Practice</i>	Functions, Events and Lists	8
<i>Animation and Games - Extended User Interaction</i> - <i>Mouse and Key Events</i> - <i>Drawing Lines</i> - <i>Crazy Ball Game</i>	Functions, Events and Lists	26
<i>Extended Array Concepts</i> - <i>Indexing Into an Array</i> - <i>Finding an Element</i> - <i>Removing an Element</i>	Functions, Events and Lists	12
<i>Extended Data Structures</i> - <i>Objects/Maps</i> - <i>Iterating Over an Object</i> - <i>Sets and Grids</i> - <i>Looping over a Grid</i>	Functions, Events and Lists	40
<i>Web Design</i> - HTML - Formatting Text - Links, Images, Lists, Tables - CSS by Tag, Class, ID	No prerequisites	75

(Supplemental) Unit 8: JavaScript Control Structures (4 weeks, 20 hours)

Learn how to use control structures such as if/else statements and loops to make more advanced programs in JavaScript.

Main Standards:

- This module contains advanced concepts on standards that were already met in the Introduction to Karel module.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9154>

Objectives / Topics Covered	<ul style="list-style-type: none"> • Booleans • For Loops • Conditionals • Nested Control Structures • While Loops
Assignments / Labs	<ul style="list-style-type: none"> • Using comparison and logical operators to control the flow of

	<p>the program</p> <ul style="list-style-type: none"> ○ Example Exercise: Inventory Write a program that keeps track of a simple inventory for a store. While there are still items left in the inventory, ask the user how many items they would like to buy. Then print out how many are left in inventory after the purchase. You should use a while loop for this problem. Make sure you catch the case where the user tries to buy more items than there are in the inventory. In that case, you should print a message to the user saying that their request isn't possible. ● Using for loops <ul style="list-style-type: none"> ○ Example Exercise: All Dice Values Write a program that prints all possible dice rolls with 2 dice To do so, you should use a double for loop. ● Drawing basic graphics using control structures <ul style="list-style-type: none"> ○ Example Exercise: Caterpillar This graphics program should draw a caterpillar. A caterpillar has NUM_CIRCLES circles. Every other circle is a different color, the even circles are red, and the odd circles are green (by even we mean when i is an even number). Use a for loop to draw the caterpillar, centered vertically in the screen. Also, be sure that the caterpillar is still drawn across the whole canvas even if the value of NUM_CIRCLES is changed.
--	--

(Supplemental) Unit 9: Control Structures Challenges (1 week, 5 hours)

In this module you'll take all the foundational concepts from the Control Structures unit to solve some programming challenges.

Main Standards:

- This module contains advanced concepts on standards that were already met in the Introduction to Karel module.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9155>

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Solving large and more complex problems that incorporate control structures
Assignments / Labs	<ul style="list-style-type: none"> ● Several challenges to tie everything learned in the module together <ul style="list-style-type: none"> ○ Example Exercise: Circles in Squares Repeatedly draw a circle inside of a square and then a square inside that circle and so on. You should repeat this while the size is greater than MIN_SIZE. Each time you draw a shape, it should be centered on the screen and a new random color. You'll need to update the size based on the size of the previous one.

(Supplemental) Unit 10: Functions, Events and Lists (4 weeks, 20 hours)

Learn to write reusable code with functions and parameters.

Main Standards:

- This module contains advanced concepts on standards that were already met in the Introduction to Karel module.

Browse the full content of this unit at <https://codehs.com/library/course/6379/module/9156>

Objectives / Topics Covered	<ul style="list-style-type: none">• Functions with and without parameters• Functions with and without return values• Nested Control Structures• Local variables and scope
Assignments / Labs	<ul style="list-style-type: none">• Using various kinds of functions such as functions with and without parameters, and functions with and without return values<ul style="list-style-type: none">○ Example Exercise: Vertical Lines Write a function that draws vertical lines on the graphics canvas. If a line is vertical, then the x-values for the endpoints are the same. The parameters to your function should be the x location, and the length, and all of your lines should start at y position 0.○ Example Exercise: Is it even? Write a function called isEven that returns a boolean of whether or not a value is even or odd. The isEven function should not print anything out or return a number. It should only take in a number and return a boolean. Once you've written this function, write a program that asks the user for integers and prints whether the number they entered is even or odd using your isEven function. You should let the user keep entering numbers until they enter the SENTINEL given.