Oklahoma Advanced Programming Syllabus

High School (165 Contact Hours)

Course Overview and Goals

The Oklahoma Advanced Programming course introduces students to computer science concepts such as internet technology, data science, artificial intelligence, and programming in Python. With a unique focus on creative problem-solving and real-world applications, students are challenged to explore how computing and technology can impact the world.

Learning Environment

The course utilizes a blended classroom approach. The content is a mix of web-based and physical activities. Each module of the course is broken down into lessons. Lessons are composed of short video tutorials, interactive learning pages, quizzes, explorations, simulations, and free-response prompts. Each module ends with a comprehensive quiz that assesses students' mastery of that module's material.

Programming Environment

Students write and run programs in the browser using the CodeHS editor. Students work with Python in the console for programming modules. They learn control structures, functions and parameters, and data structures. Students then expand their use of Python with a data science focus.

More Information

Browse the content of this course at https://codehs.com/course/24142/overview

Prerequisites

The Oklahoma Advanced Programming course is highly visual, dynamic, and interactive, making it engaging and accessible for those new to computer science.

Course Breakdown

Module 1: Programming with Python (2 weeks/10 hours)

In this module, students build on the basics of Python and learn user input, arithmetic expressions, mouse events, and how to create graphics.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33222

Topics Covered	• Variable Names • Data Types • Arithmetic Expressions • Order of Operations • Modulus • String Concatenation • User Input

	• Program Output • Using Python Graphics • Mouse Events

Example Assignments	• Printing
	  ○ Print messages to the console
	• Variables
	  ○ Create variables of different types, and print them to the console.
	• Types
	  ○ Investigate the types of different variables
	  ○ Convert between types
	• Graphics
	  ○ Students will create a snowman by creating circles of different sizes.
	• Mouse Events
	  ○ Students will create a program that draws a rectangle whenever they click the mouse.

Module 2: Python Control Structures (2 weeks/10 hours)

In this module, students learn how to use booleans and logical operators with control structures to create more advanced programs in Python.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33223

Topics Covered	• Booleans • Relational Operators • Selection • Conditional Statements • Nested Conditionals • Random Numbers • Iteration • Loops

Example Assignments	• Comparison Operators
	  ○ Students will write a program that stores three variables from the user: their points per game, their rebounds per game, and their assists per game, in that order.
	• While Loops
	  ○ *Better Password Prompt:* Students write a program that uses a while loop to prompt a user for a password. They keep prompting the user for the password, and if they get it correct, they then break out of the loop. If they don't get it correct, they should give the user an error message. This activity requires that students use multiple program statements in a specific order to solve a problem.
	• Challenge Activity
	  ○ The computer picks a number between 1 and 100, and you have to guess it. The computer will tell you whether your guess was too high, too low, or correct. Your assignment is to generate a random number and let the user guess numbers until they guess the correct number.

Module 3: Functions and Parameters (2 weeks/10 hours)

In this module, students will build on their knowledge of functions and learn to write reusable code using functions and parameters.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33224

Topics Covered	• Program Output • Procedures • Parameters • Return Values • Using Existing Algorithms

Example Assignments	• Functions and Return Values
	  ○ *Pool Table:* Students write a program with a function that draws a pool ball. This function should take as parameters, the color, the number that should go on the pool ball, and the location of the center of the pool ball. Students need to consider the function abstractly as a means for taking specific data via the parameters and creating a unique graphical output based on those inputs.
	• Challenge Activity
	  ○ Write a program to draw ghosts on the screen. You must do this by writing a function called `draw_ghost`, which takes three parameters, the x location of the ghost, the y location of the ghost, and the color of the ghost. x and y for the ghost define where the center of the head should go.

Module 4: Strings (3 weeks/15 hours)

In this module, students use more sophisticated strategies for manipulating text in their programs - slicing, concatenating, and formatting.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33252

Topics Covered	• Indexing and Slicing • Math Operators on Strings • For Loops Over a String • String Methods

Example Assignments	• Example exercises:
	  ○ Indexing
	  ■ First character - write a function that takes a string and returns the first character
	  ■ All but the first character - write a function that takes a string and returns everything but the first character
	  ○ Math operators and strings
	  ■ Full name - write a function that takes two strings (a first name and a last name) and returns a full name as a single string
	  ■ Replace a letter - write a function that takes a string and returns a copy with the character at a particular index replaced with a dash
	  ○ For loops on strings
	  ■ Count occurrences - write a function that takes two strings and returns the number of times the second string appears

	in the first string ○ String methods ■ Add enthusiasm - write a function that takes a string and returns that string in all upper-case ■ Remove all from string - write a function that takes two strings and returns a string that consists of the first string with all instances of the second string removed

Module 5: Basic Data Structures (2 weeks/10 hours)

In this module, students learn the basics of lists in Python while exploring tuples and lists. Students will explore how to create each of these data structures and the various methods to access or alter them.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33226

Topics Covered	● Data Values ● Lists and Elements ● Indices ● List Procedures ● Traversing a List ● Iteration Statements ● Using Existing Algorithms
Example Assignments	● For Loops and Lists ○ *Owls*: Your friend really likes talking about owls. Write a function `owl_count` that takes a block of text and counts how many words they say have the word "owl" in them. Any word with "owl" in it should count, so "owls," "owlette," and "howl" should all count. ● List Methods ○ *Librarian, Part 2*: Students write a program to ask the user for an author's full name. Students will then use list procedures to split the full name into individual names and then slice the list to add the last name to a new list. Once the students have collected all of the last names, they will sort them and then print the results. This program development requires students to use user input data that can contain a variable number of names. The students must then use various list techniques to manipulate the data.

Module 6: Digital Information (2-3 weeks/10-15 hours)

Students learn about the various ways we represent information digitally including number systems, encoding data, and creating pixel images.

Browse the full content of this unit at https://codehs.com/course/24142/explore/module/33227

Topics Covered	● How digital data is represented ● Encoding data ● Converting and using different number systems ○ Binary ○ Hexadecimal ● Manipulating images at the pixel level

Example Assignments	● Encoding data ○ Create your own encoding scheme ○ Encode images using binary ○ Example Activity: ■ Write a message by encoding the characters in binary, using the ASCII codes. ● Using different number systems ○ Convert numbers between decimal, binary, and hexadecimal ○ Example Activity: ■ Earn a high score playing the decimal to binary game or decimal to hexadecimal game. Click on the digits to change their values and make the binary or hexadecimal number match the target decimal value. ● Manipulating Images ○ Make different colors by changing the amount of red, green, and blue present ○ Create image filters ○ Create images pixel by pixel ○ Example Activity: ■ Create a pixel rainbow by typing out the hexadecimal color encoding for each pixel

Module 7: The Internet (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/24142/explore/module/33230

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 ● Impact of the Internet
Example Assignments / Labs	● Introduction to the Internet ○ What is the Internet? How does it work? What has been its impact on society? ○ Why do we need protocols for the Internet? ○ Example Activity ■ Explore the different levels of the internet. ● Decimal to Binary ● Hexadecimal ● Bits to ASCII ○ Hello World in Bits

	• Internet hardware
	  ◦ Vocabulary: bandwidth, bitrate, latency
	  ◦ Why are protocols so important?
	  ◦ How do we send data over the Internet?
	  ◦ Example Activities
	  ▪ 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
	  ◦ Vocabulary: Internet Protocol (IP)
	  ◦ How do IP addresses compare to postal addresses?
	  ◦ How IP addresses work?
	  ◦ Example Activities
	  ▪ Explore the differences between IPv4 and IPv6. Why are we running out of addresses?
	  ▪ Trace a website request from the server, through the network, and to your computer
	• Domain Name System (DNS)
	  ◦ How does DNS help with sending digital information and IP addresses?
	  ◦ Example Activities
	  ▪ 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
	  ◦ How data is transmitted?
	  ◦ How are internet packets able to find their way to your computer?
	  ◦ Example Activities:
	  ▪ 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.
	  ◦ What are the standard protocols for the Internet and how do they work? (TCP/IP, HTTP)
	• The Internet and Cybersecurity
	  ◦ What are cybercrime and cyberwarfare?
	  ◦ How do we network attacks? (certificate authorities, public key encryption)

Module 8: Computing System (1-2 weeks/6 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/24142/explore/module/33421

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

Example Assignments / Labs	• 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 9: Introduction to Data Science (4 weeks/20 hours)

Students will learn and apply the process of the data science life cycle. This includes asking statistical questions, collecting or obtaining reliable raw data, analyzing the data using measures of central tendency and spread and interpreting, and summarizing the results.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33271

Topics Covered	• What is Data Science?
	• Gathering Data
	  ○ Quantitative/Qualitative
	• Exploring Data Using Python
	• Modules and Libraries
	• Using the Pandas Library
	  ○ Series
	  ■ Measures of Central Tendency
	  ■ Measures of Spread
	  ○ DataFrames
	  ■ Selecting Columns
	  ■ Using Functions

Example Assignments	• Mini-Project:
	  ○ Students will go through the first two steps of the data cycle using data of their choosing.
	• Hot Dog Plots:
	  ○ Use the correct Python functions to create a boxplot of the data. Using the graph, determine the summary statistics and the spread.

	• Roller Coaster Rankings:
	  ○ Define a function that will compute a score for each roller coaster. Use this function to store the results in a new column.
	• Student Test Scores:
	  ○ Create a function that finds the maximum test score between test one and test two for each student. Create a function that finds the maximum test score between all three tests for each student. Decide which calculations, along with these two new columns, can help you answer the original statistical question. Explore and further analyze your data until you come to a conclusion.

Module 10: Data Science for Business (5 weeks/25 hours)

Students will gather business data that can be used to make decisions about how to better the company or product. They will present their findings in a business report that suggests several action items that they predict will help the business's performance and growth.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33274

Topics Covered	• Determining Dataset Quality
	• Aggregating Data
	  ○ Grouping
	  ○ Sorting
	• Combining Datasets
	  ○ Concatenating
	  ○ Joining/Merging
	• Bias in Data Analytics

Example Assignments	• Project - Data Science for Business: Students will gather (or create) business data that can be used to make decisions about how to better the company or product. They will present their findings in a business report that suggests several action items that they predict will help the business's performance and growth. • Determining Completeness: The raw datasets (without any cleaning) are provided. Your task is to check the completeness of at least two different datasets to compare and contrast them. • Fuzzy Book Titles: Use the FuzzyWuzzy library to extract all titles that are close to matching the word "batman". Print out the results. Are they unique book titles or do you notice errors in the titles? • Billionaire Sort: Group the dataset by gender. Only display the gender count. What insights does this grouping give you? Group the dataset by industry. Only display the industry count. What insights does this grouping give you? Sort by age and then net worth. You will want the oldest person to be listed first. If there are two people of the same age, you'll want the person with the highest net worth to be listed first.

Module 11: AI in Gaming (2-3 weeks/15 hours)

In this module, students learn how AI has been used in gaming and other applications, and create an unbeatable computer Tic Tac Toe player!

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33255

Topics Covered	• Human & Artificial Intelligence • Artificial Intelligence in Gaming • Finite States • Search Trees • Creating an Non Player Character • Search Trees and Recursion • Minimax • Depth and Alpha Beta Pruning

Example Assignments	• Building Tic Tac Toe:
	  ○ Students will build a game of Tic Tac Toe, where the computer isn't learning or strategizing how to play the game, but rather is being programmed directly to do so.
	• Implementing Minimax
	  ○ Students are introduced to the recursive function minimax, which allows a computer to search through all possible moves in a game and pick the most optimal move. Students then implement this function in their existing version of Tic Tac Toe to create an unbeatable version of the game.
	• Implementing Depth
	  ○ Students consider the purpose of optimization, and whether game play should include such measures. Students then learn how to limit the depth of search that minimax makes, making it so that the game can be beaten if a player makes optimal moves. Students then implement depth search in their existing Tic Tac Toe game.

Module 12: Software Development (2 weeks/10 hours)

In this module, students will learn and use the software development life cycle to create a simple app that will solve an everyday problem.

Browse the full content of this module at https://codehs.com/course/24142/explore/module/33470

Topics Covered	• Software Development Life Cycle • Prototyping • Using Flowcharts • User Testing • Software Licenses

Example Assignments	• Example Exercise: Create a Flowchart
	  ○ Create a flowchart that visualizes the flow of your app. You can use paper or a digital tool to create your flowchart. This flowchart will still be a draft - you can always come back and edit it later on as you continue development!
	• Example Exercise: Test Your App!
	  ○ Now that you have a plan, you are ready to conduct your user testing! You should have at least two different users test your app. In

	the editor, submit the data you collect during testing. This could be a link to images, videos, or a text document, or a summary of your findings typed into the editor.