FUNDAMENTALS: UNIT PLAN

girls who code
SUMMER IMMERSION PROGRAM

In this unit, students will begin to grapple with the essential questions below through an introduction to Computer Science. By the end of the week, students will be able to use and explain the Core4 of computer programming in order to solve problems. GWC defines the Core4 as variables, loops, conditionals, and functions.

Students will learn about these concepts through two different languages: Scratch, a block-based visual programming language, and Python. They will begin by exploring these concepts in Scratch, and will show their developing understanding of these concepts by creating a side-scrolling game. Scratch is a great entry point for students to begin grappling with these abstract concepts because they can get straight to creation!

Then, students will reinforce their understanding of these concepts by learning Python. Students will show their deeper understanding of the fundamental concepts by comparing the use of the Core4 in Scratch and Python. By the end of the unit, students should have a firm grasp on these fundamental concepts and be prepared to see how these concepts work in applications (unit 2).

Unit Essential Questions	Unit Enduring Understandings
Who are Computer Scientists?	GWC girls are computer scientists! Computer scientists come from all different backgrounds. Diversity is important in CS because a diverse set of voices and perspectives improve us as people, as well as the products we create. Women were some of the earliest software designers.
What is Computer Science? **What tools do computer scientists use to solve problems?**	Computer science is the study of solving problems using computational thinking, or by taking the steps needed to solve a problem and turning it into something a computer can follow. The Core4 are fundamental tools in computational thinking. Loops, Variables, Conditionals and functions are all ways that humans can communicate solutions to computers.
How and why do we use computer science to solve problems?	Computer science is more than building programs - "doing" computer science requires problem solving. Have plan on paper before trying to code. You hardly ever get it working the first time, but keep trying!

How and why do we work with people to solve problems? **How do I work best?** **How can I communicate effectively with my teachers and peers?**	Working with different kinds of people can be hard, but it pushes you to think outside of your comfort zone, create better projects, and become a better person.

STUDENT OUTCOMES

Classroom Culture Goals:

- Students have a clear understanding of classroom procedures and host-site expectations.
- Students have built trust with an adult by having at least one 1:1 check-in with someone on the teaching staff.
- Students have meaningfully reflected on their developing understanding through journalling.
- Students have discussed at least one woman leader in technology.

Skill Goals	Knowledge Goals:
Students will **be able to:** • Use variables, loops, conditionals and functions in both Scratch and Python to solve problems. • Articulate the similarities and differences between Scratch and Python. • Give at least two examples of Women in computer science.	Students will **know:** • How to meet classroom and host-site expectations through clear procedures. • How to explain variables, loops, conditionals and functions are to another student. • How to define computer science in their own words.

KEY EVIDENCE OF SUCCESS

Performance Task:

- Drawing shapes compare contrast
    A. Present about similarities and differences in drawing shapes using Python and Scratch

LEARNING PLAN

WEEK 1: WELCOME AND SCRATCH

Lesson Plan and Overview	Daily Goal	Outputs and Look-Fors
Monday: Welcome **Overview:** The main goal of today is to start building a positive classroom culture! Students will get to know each other and the teaching team through a number of ice-breaker and bonding activities. After breaking the ice, students will outline the rules they will hold each other accountable to by creating their Classroom Contract. Girls will also orient themselves to the Scratch programming environment and create their first project.	**Students understand:** • They are part of a supportive community. • What we expect of them throughout the summer. • They are active members of their classroom community. **Students can:** • Log in to Scratch. • Use blocks to make a sprite do something.	**Classroom Contract** • All girls are participating. **Scratch Welcome Project** • Girls are using a variety of blocks in a creative way.
Tuesday: Draw Shapes **Overview:** The primarily goal of today is for girls to begin to see how they can use the structure of one problem to effectively solve other similar problems. The progression of the day leads girls to look at the similarities between what is required to draw a square, a triangle, and finally any shape.	**Students can:** • Use a loop in a scratch program to repeat a task. • Use user input to change the output of a scratch program. • Explain their program step by step.	**Draw a Square** • Girls can explain how the "Repeat" block works. **Draw a Triangle** • Girls can extend their original code to a new scenario. **Draw Any Shape** • Girls can use variables to make their code responsive.

Wednesday: Paddle Ball **Overview:** The main goal for today is for girls to be able to give examples of how conditionals and custom blocks can be programmatically useful.	**Students can:** • Use conditionals in a Scratch program to change a sprite's behavior. • Create an animation in Scratch. • Create a custom block in Scratch. • Compare the concept of function in math to that in Scratch.	**Paddle Ball** • Girls use conditionals in their game. • Girls create a custom function block and can explain why it is useful.
Thursday: Side Scroller Game **Overview:** Girls begin the day by exploring some popular games that they may be familiar with in order to see that many of them are classified as "side scrolling" games. With these prototypes in mind, they will collaborate to take their game from idea to functionality by storyboarding, doing graphic design work, and actually coding their game in Scratch	**Students can:** • Use the Core4 correctly in a project with a broader scope. • Use the clone feature in Scratch. • Explain how and why cloning is useful.	**Side Scroller Game** • Girls are collaborating and compromising as they design their game. • Girls are developing programs that effectively use the Core4 Concepts.

WEEK 2 - PYTHON

Lesson Plan and Overview	Daily Goal	Outputs and Look-Fors
Monday: Intro to Portfolio Web-page **Overview:** This is a sentence that will tell the story of the day.	**Students can:** • Edit HTML to customize a web page. • Explain what an HTML tag is and use one correctly. • Embed an image in an HTML page. • Fork a github repository.	**Intro to Portfolio Web-page** Girls are using internet resources and persever-ance to set up their first website.
Tuesday: Draw Shapes in Python **Overview:** The goal for today is for girls to begin to generalize the concepts they have learned in Scratch to the text-based language of Python by replicating a project from week 1 in Python. As they do this, girls will also learn how to use command-line code to run simple programs	**Students can:** • Run a python program. • Use loops and variables in a Python Program. • Define syntax. • Access terminal. **Students know:** • How Scratch and Python compare to one another.	**Draw Shapes in Py-thon** • Girls can explain how their project is similar to the code they used when coding in Scratch.

Wednesday: Text Adventures and Core4 Portfolio Pages **Overview:** Students begin the day by exploring a simple, story-driven game that takes user input. They will then design their own story that they will code in Python, as they learn the new skill of using conditionals and nested conditionals. After completing their Text Adventure and sharing the results, girls take some independent reflection time to explain - in their own words - the meaning of the CS concepts they've learned so far by building a Core4 Portfolio page on their website and sharing it with other students.	**Students can:** • Use conditionals and nested conditionals in a python program. • Use edge cases to test conditionals. • Define and give examples of loops, variables, conditionals and functions.	**Text Adventures** • Girls have created a program that accurately uses nested conditionals for program control. • Girls have tested cases to ensure their program can handle incorrect input from a user. **Core4 Presentations** • Girls are able to provide a definition of each of the Core4 concepts. • Girls are able to add this content to their webpage.
Thursday: Obamicon **Overview:** In this culminating lesson, girls are challenged to work collaboratively to create a program that leveraged more advanced data structures and image libraries. The understanding of a list is necessary for girls to master before moving into application of these key concepts in the following two weeks.	**Students can:** • Explain how to access specific items within a list. • Use multi-dimensional lists to store and manipulate data, including adding items to lists. • Explain the purpose of the Pillow Python Library and accurately use functions from the library. **Students know:** • That libraries and their documentation can be useful. • How to visualize multi-dimensional arrays.	**Obamicon** • Girls can loop through lists. • Girls are able to create functions.

WEEK 1: SUGGESTED CALENDAR

Monday: Welcome	Tuesday: Draw Shapes	Wednesday: Paddle Ball	Thursday: Side Scroller
• Classroom Contract - 60 min • Ice breaker - 30 min	• Introduction - 30 min • Draw a Square - 60 min • Draw Triangles - 30 min • Suggested - Break • Abstracting - 30 min • Introduce New Information - 30 min • Suggested - Lunch • Finish Project - 45 min • • Presentations - 30 min • Reflection - 15 min • Wrap-up - 15 min	• Introduction - 15 min • Introduction - What is a Conditional? - 15 min • Mini-Coding Challenge - 25 min • Suggested - Break • Introducing Paddleball - 20 min • Workblock 1 - 35 min • Suggested - Lunch • Suggested - Bonding • Introduce New Information - Functions • in Scratch - 30 min • Finish Project - 90 min • Presentations - 20 min • Reflection - 20 min • Wrap-up - 20 min	• Introduction - 40 min • Storyboarding - 45 min • Suggested - Break • Graphic Design - 45 min • Suggested - Lunch • Gameplay Design - 45 min • Cloning in Scratch - 45 min • Code Brainstorm - 15 min • Finish Project - 45 min • Suggested - Break • Presentations - 15 min

WEEK 2: SUGGESTED CALENDAR

Monday: Portfolio Website

- Introduction - 20 min
- Women In Tech Spotlight - 20 min
- Accessibility - 20 min
- Start the Project: Setting Up GitHub Pages - 20 min
- Suggested - Break
- Wireframing - 40 min
- Suggested - Lunch
- Work block 1 - 45 min
- Gallery Walk - 15 min
- Work block 2 - 60 min
- Presentations - 30 min
- Reflection - 15 min
- Wrap-Up - 15 min

Tuesday: Draw Shapes - Python

- Introduction - 30 min
- Terminal Tutorial - 40 min
- Suggested - Break
- Introduction to Python Syntax and Datatypes - 45 min
- Suggested - Lunch
- Introduction to Notepad and Loops in Python - 10 min
- Draw-Shapes - Paired Coding - 45 min
- Code Brainstorm - 15
- Finish Project - 60 min
- Suggested - Break
- Presentations - 15 min
- Reflection - 15 min
- Wrap-Up - 15 min

Wednesday: Text Adventure

- Introduction - 20 min
- Text Adventure Intro - Designing Your Story - 45 min
- Suggested - Break
- Conditionals and User Input in Python - 45 min
- Finish Project - 60 min
- Suggested - Lunch
- Presentations - 20 min
- Suggested - Break
- Suggested - Bonding
- Wrap-up: 15 mins

Thursday: Obamicon

-
- Introduction - 30 min
- Intro - How do computers store images? - 20 min
- Suggested - Break
- Introduction to Data Structures - 110 min
- Suggested - Lunch
- Combining Core4 with Lists - 60 min
- Introduction to the PIL Library - 25 min
- Finish Project - 60 min
- Suggested - Break
- Presentations - 10 min
- Reflection - 15 min
- Wrap-Up - 15 min