FIRIA LABS

LESSON: May Flowers Project		Time: 45- 60 minutes
Overview: This lesson will display May flowers as the May monthly theme. It is the second part of the April Showers project.		**Objectives:** • I can use circles to draw a flower. • I can use nested for loops to draw several flowers on the display. • I can select a random color from a list.
Grades 6-8 CS Standards: **2-CS-01** Design projects that combine hardware and software components to collect and exchange data. **2-CS-03** Systematically identify and fix problems with computing devices and their components. **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation and review of programs. **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. **2-AP-16** Incorporate existing code, media and libraries into original programs, and give attribution.	**Grades 9-10 CS Standards:** **3A-DA-11** Create interactive data visualizations using software tools to help others better understand real-world phenomena. **3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. **3A-AP-16** Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. **3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules and/or objects.	**Grades 11-12 CS Standards:** **3B-AP-10** Use and adapt classic algorithms to solve computational problems. **3B-AP-14** Construct solutions to problems using student-created components, such as procedures, modules and/or objects. **3B-AP-16** Demonstrate code reuse by creating programming solutions using libraries and APIs. **3B-AP-17** Plan and develop programs for broad audiences using a software life cycle process. **3B-AP-22** Modify an existing program to add additional functionality and discuss intended and unintended implications.

Preparation:
• Download slides
• Be familiar with the final code
• Read through the teaching guide
	In the folder:
	• May Showers project slides
	• May Showers project code
	  ○ Starter code
	  ○ Code solutions for all steps and final
		Agenda:
		• Complete program using slides (45-60 minutes)

Teacher Notes: • This lesson is designed so that students can work independently by following the slides. However, you can also work together as a class by projecting the slides. See the teaching guide for specific help and hints. • Almost all mistakes made by students are typing mistakes. If students get errors when they run their code, first look over the code for spelling, punctuation and indenting. • The solution code for each step is provided. • The project has two parts – April and May. May Flowers is part 2. If your students haven't completed the April project (part 1), you can either give them the starter code for May, or have them complete the April project first.		

Teaching Guide

Warm-up

A warm-up isn't really needed for this lesson. You can review the April project, review concepts in Python or discuss debugging techniques. Or, just start the lesson.

Create/Run the Program (~45 minutes)

💻 This project can be completed individually or with pair programming.
It can be completed working independently or as a whole class.

💡 **Teaching tip:**

This project is not included in CodeSpace. Download and follow the slides. They include step-by-step instructions as well as code snippets to guide students through the program code creation.

You can have students complete the project one of two ways:

- Show the slides on a large screen or monitor and have the class work on each step together.
- Give the slides to the students and let them work through the instructions at their own pace. This works well if students are pair programming and have access to one computer for instructions and one computer for programming.

💻 **Slide 3 Open Project File**

Students get into the sandbox of CodeSpace and open the April_Showers project! If students do not have the April_Showers program file, you can either have them do part 1 first, or give them starter code for this project, which is the final file for the April project.

💻 **Slides 4-7 Step #1 Create a flower**

Students add a function that draws a single flower using filled-in circles.

- Sample code is given on the slides. However, if students want to create their own flower, that is fine. Just make sure it isn't very big so you can have multiple flowers on the screen.
- The function will use two parameters: x and y. These are used like (x, y) coordinates for the location of each circle.
- Another parameter is used for the color: col. It will be utilized in a future step.
- IMPORTANT! When students add test code to see their flower, they should not delete the code that is already in the Main Program; they will still need it. Just add the test code at the top of the Main Program.

💻 **Slides 8-12 Step #2 Add the function to display 6 flowers**

Students add another function that uses for loops to display two rows of flowers.

- The code given is for six flowers: two rows of three flowers each. If students know how to get random numbers, they can do so as an alternative. Get a random number for x and a random number for y. Use them when calling the flower() function. If students use the alternative, only one for loop is needed.

💻 **Slides 13-14 Step #3 Add text and ground**

Students add code to may_flowers() to label what is displayed and show a ground.

- If students are following the slides and displaying six flowers, they should keep this step as indicated on the slides.
- If students use the alternative random flowers, they may not want the rectangle ground. Also, they can move the text to wherever they want.

💻 **Slides 15-19 Step #4 Colorful flowers**

Students add a list for flower colors, and then use the list to select colors for their flowers.

- The list can have any number of colors. The one shown in the sample has six colors, but really it can be any number, and they can choose any colors.
- If students know how to do RGB triplets, they can be included in the list.

💻 **Slides 20-25 Step #5 Make it interactive**

This is the final step. Students will write code for the main program, which brings all the functions together.

- Students add an if statement with a button press for displaying the May flowers.
- Flowers should only appear after rain, so add another Boolean variable to indicate rain. This part can be skipped if you want it to display flowers any time, not just after rain.
- Another alternative is to set the rained Boolean variable back to False in the second button, so it must rain before flowers appear. The example code enables flowers to appear anytime after the first rain.
- Another alternative, if programming random flowers, is to use two button presses, one for rows and one for random flowers. Sample code for this alternative is included.

💡 **Teaching tip:**

This is a good program for a remix. Challenge students to add or change something so their program is different from other students' programs.

Wrap-up

💻 You can wrap-up this project in a variety of ways, depending on your students and your classroom procedures.

- Students can share their projects with other students, especially students not in class. Challenge them to do this!
- Students can fill out a journal entry about their experience or what they learned during the lesson.