Course: West Virginia Discovering Computer Science | Module: Exploring Code with Karel

Lesson 1.4: Multiple Functions

https://codehs.com/course/20502/lesson/1.4

Description	In this lesson, students learn in more detail about functions and how to use functions to break down their programs into smaller pieces. Students will also learn about using the main function and commenting code to make it easier to understand.
Objective	Students will be able to: • Explain the difference between defining and calling a function • Utilize functions to write higher level Karel programs • Break a large problem down into smaller, simpler problems • Create clear and readable comments in their code that help the reader understand the code
Activities	1.4.1 Video: Top Down Design and Decomposition 1.4.2 Check for Understanding: Top Down Design and Decomposition Quiz 1.4.3 Video: Top Down Design and Decomposition in Karel 1.4.4 Example: Using the Main Function 1.4.5 Exercise: Tennis Ball Stacks 1.4.6 Example: Comments in Karel 1.4.7 Debugging: Clean Up
Prior Knowledge	• Basic syntax rules in writing commands and functions. • How to create a functions in Karel
Planning Notes	• Review the slides for the lesson before the start of class • Consider telling a story about a project that took a considerable amount of time, and the easiest strategy used to solve it. • Revisit student created functions from earlier if needed. • Speakers that enable students to hear music are needed for the add-on handout activity. • There are handouts associated with this lesson. Consider using them in class, as an exit ticket, or as homework.
Standards Addressed	

Teaching and Learning Strategies

Lesson Opener:

- Review material from the previous day of class using discussion questions. Have students as a warm up write a function that gets Karel to put `threeBalls()` on Karel world space. Have students share what they wrote with a partner, and allow one student to write their answer on the board. Explain that this lesson will teach students a bit more about functions. [5-10 min]
    - Alternatively, use the *Dancing with Functions* handout to have students write out functions for real life problems.
- Have students complete *Decomposition* handout. Students should share with the class the solutions they come up with. [5-8 min]
    - Challenge students to be very specific when completing the handout. Rather than tell the computer butler to make a sandwich, have students consider the most basic command they would have to tell the computer in order to get it to spread jelly on a piece of bread.
- Provide students with a mystery function that uses uncommon English words as function names. Have them attempt to decipher what the function does. Ask students to consider what kind of actions the creator of the function could do to help others use their function, even if they kept the sophisticated language. Explain that comments can be used in coding to help others understand the code that we write. [10 min]
    - Refer to discussions questions for assistance.
    - Here is a sample mystery function:

```
function mystery() {
   girate();
   traverse();
   ascend();
   retract();
}
```

Activities:

- Watch *Top Down Design and Decomposition* video and complete the corresponding quiz. [6-8 min]
- Watch *Top Down Design and Decomposition in Karel* video. [5min]
- Review the *Using the Main Function* example program.
    - Discuss the importance of the `main()` function and how it's called and defined.
    - Have students complete *The Main Function - My Day in Functions* handout.
- Complete the *Tennis Ball Stacks* exercise [5-10 min]
- Review the *Comments in Karel* example program.
- Complete the *Clean Up* debugging exercise.
    - Encourage students to use the comments in the program to help them debug the program.

Lesson Closer:

- If there is time at the end of class, have students reflect and discuss their responses to the end of class discussion questions. [5

	min] • Use handouts as exit tickets or homework if they were not used during class. [5 min]

Discussion Questions	**Beginning of class:**
	• What is a function?
	  ◦ *A function is a way of teaching Karel a new command using preexisting commands*
	• How do we go about creating functions in Karel?
	  ◦ *Based on what students know, functions need to be given names, using () at the end, and function at the beginning. The commands given to that function are enclosed in { }.*
	• Think about a big problem or big project you might encounter. How might you go about solving the problem, or starting the project?
	  ◦ *Student could create a flowchart of how to break problem down, make a list of steps to complete a task, etc.*
	• What would make it easier to understand the `mystery` function?
	  ◦ *An accompanying document that explains how it works, an explanation from the user, a dictionary, etc.*
	• What do you think the `mystery` function does?
	  ◦ *It has Karel dance, move forward, go up, and then come back.*
	
	End of class:
	• What does it mean to define a function?
	  ◦ *Defining a function is to give the method a series of commands. Whenever the function is called, it will execute the commands in the definition.*
	• What does it mean to call a function?
	  ◦ *Calling a function means to use the function in one's code. If the function is defined without being called, the commands of the function will not be executed.*
	• Describe how you helped Karel break down a big problem into a smaller problem today.
	• How did that affect how you wrote your code?
	  ◦ *Answers will vary. Students should say that it made them more likely to create additional functions.*
	• How does the main function make your programs easier to understand?
	  ◦ *The main function makes programs easier to understand because it consolidates all of the actual commands that are going to be executed in the program into one place, which makes the program more readable.*
	• Why is it important to use comments in your programs?
	  ◦ *It helps with the readability of your code, and allows others to figure out how your programs work!*

Resources/Handouts	Commenting Code (Student)

	Commenting Code (Teacher) Decomposition (Student) Decomposition (Teacher) Planning Programs (Student) Planning Programs (Teacher) The Main Function - My Day in Functions

Vocabulary

Term	Definition
Comment	A message in your code that explains what is going on.
Comment Out	Commenting out code makes the computer ignore it, so it does not run.
Decomposition	Decomposition is breaking your program into smaller parts.
Top Down Design	Top down design is a method for breaking our program down into smaller parts.

Modification: Advanced	Modification: Special Education	Modification: English Language Learners
• Have students create an original Sandbox program that defines and calls a minimum of 4 (four) functions. • Have students decompose more complicated tasks from their every day life. • Have students decompose the same problem into multiple solutions with various levels of granularity.	• Use a printed list of Karel's commands from the DOCS tab and have student circle which commands they will use to define the function. • Pair programming with another student • Print out video slides for students to reference • Have students complete the "Planning Programs* and *Commenting Code* handouts instead of some of the exercises.	• Complete a flowchart that diagrams how a function works in a program. • Print out video slides and have students use dictionary to translate unknown vocabulary • Have students complete the "Planning Programs* and *Commenting Code* handouts instead of some of the exercises. • Pair programming with another student • Exit ticket: What is the difference between defining a function and calling a function?