L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

Directions for sharing

- Please change the document name to *L05_container_variables_yourusernames*.
- e.g. *L05_container_variables_pearcej_wilbornew*
- One person should make a single copy of this worksheet in their Google account.
- Then that person should share it with everyone in the group so they can all edit it. Note that having access will be essential for studying for exams.
- You will be submitting both a downloaded pdf of this document and a program when you are done.

Member Roles

- If you have three people be sure to rotate through all three instead of using odd/even.

Team Roles	**Member Name**
First Driver "Drives" at the keyboard on all odd levels, listening to partners. **Facilitator:** Keep track of time and make sure every player contributes and learns equally.	
Second Driver "Drive" at the keyboard on all even levels, listening to partners. **Reflector:** Consider how the answers could be deeper, and how the team could work and learn more effectively so everyone learns	
Third member: Drives every third time.	

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

Actively comments on all ideas and code.	

Lab Activity:

Objectives

- Learn to use "containers" variables in RoboLab
- Learn to use the RoboLab Help features

Tools and Parts Needed

- Constructed 2-wheeled tail-dragger robot
- A touch sensor, embedded in a "bumper"

What are Variables and Containers?

Variables are very important in programming. Variables enable us to store information for future use, and the power of using collected information to direct the running of a program is part of what makes programming so powerful. By using variables, the program becomes more flexible--instead of having only constants in a program, variables allow the program to store data that might vary from one run to another.

For example, in algebra **x** is a variable and **x = 2** means 2 is assigned to variable **x**, so in many programming languages, we have a statement such as:

x = 2; // which is called an assignment statement because it assigns the value 2 to the variable named x, and the value may change in the program.

In programming, variables are used to remember information for later use.

1. **(5 min)** Give an example of a piece of information that you think you might want a robot to remember that might be able to be stored in a variable.

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

Variables in RoboLab

Variables in RoboLab are called containers and are found under the container icon:

There are three different colors of general-purpose containers: red, blue and yellow:

 Each different color represents a different general-purpose container, and the colors are modifiers, so are found in the modifiers area. Beyond the colors, there are actually 48 generic container variables.

Containers are the color of the jars whereas the **container value** is the value stored in it and is accessed icons as well: 

Each container is something called a **global variable** because it can be used everywhere in the program. A container can hold one value at a time and it can only hold only an **integer (whole) number**. Therefore, it cannot hold a real or floating point number like 3.14. (Actually, this seems like a more major restriction than it really is.) One thing that may be hard to understand is that, containers always have a value in them, even before you store one, so they must always be initialized or set to an appropriate starting value like 0. (Otherwise, they may have a "garbage" value in them.)

Containers in RoboLab work a bit like the memory on a calculator. To clarify how to use a RoboLab container, you need to understand how to initialize a container, how to store a value in a container, and how to access the value in the container.

2. **(5 min) As mentioned above, there are 3 different colors of containers in RoboLab:**

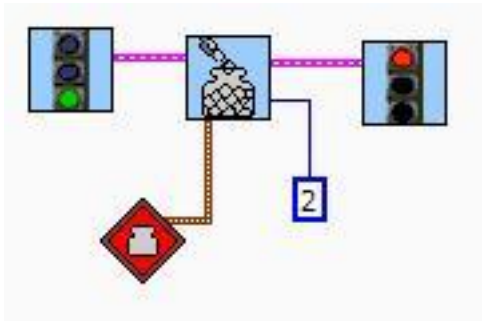
   Open RoboLab and determine which one of these colors is the default color (This means it will use this container if you don't use a color modifier.) Explain how you know this is the default color. Don't just guess or assume find someplace where RoboLab tells you which is the default container.

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

Choosing the container has a different icon from accessing a specific container's value. For example:

	Red container icon: Used to indicated **which container** to use.
	Red container value icon: Used when you want to **use the value that was already stored** in a particular container.

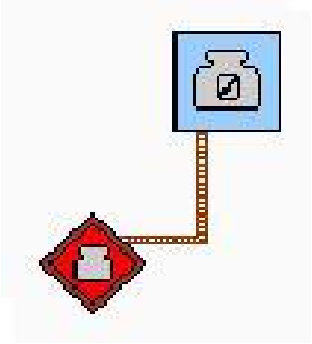
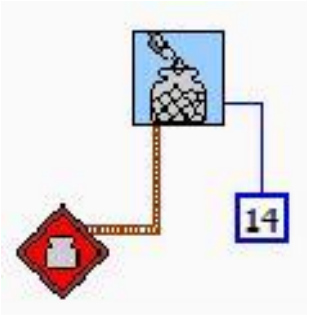
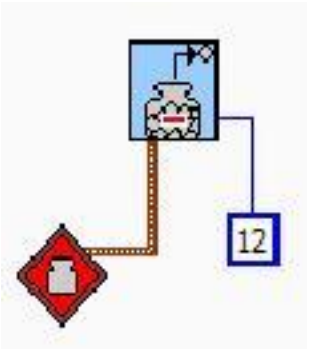
	This is called an **assignment statement** because 2 is assigned to (stored in) the red container variable.

3. **(2 min)** The following two icons are used in different situations:  In your own words, describe when you would use each of these.

Some examples of using container variables in your program:

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

In most programming languages, variables should be initialized (to zero of some other initial value) before one uses them. In RoboLab, the initialization of a container basically sets the value of the container to zero so that the container will function correctly. Any container (Red, Blue or Yellow) should be set to zero value before employing it in your program. In the diagram, the value of a red container is set to zero.	
Now after initialization, a programmer uses the variable to store data or any information by assigning values. In RoboLab, we store data in container by filling it with the data. Here in the diagram, the red container is set to the value of 14.	
Simple math can be used to store data and change the values that are already stored. There are other mathematical operations like subtraction, division, and addition. So one can use any math and also any whole number value to use with those operations. In the diagram on the right, the number 12 is being subtracted from the red container. Therefore, if this is run after the above command, the value of the red container is now 2. (14 - 12 = 2)	

L05: CSC 126, Introduction to Robotics

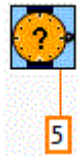
This is a team assignment designed as an in-class activity.

Waiting

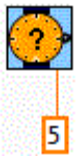
We have seen that it is possible for the robot to wait for an amount of time:

Later we will see the robot wait for darker, lighter, dark, and light:  ,  ,"  , and  using the light sensor.

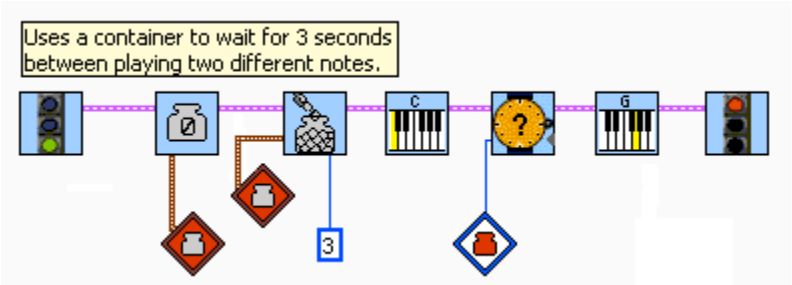
There are other things the robot can be asked to wait for. For example, we might ask the robot to wait for an amount of time other than one of the defaults by using the following:



4. **(5 min) Given that the following code:**

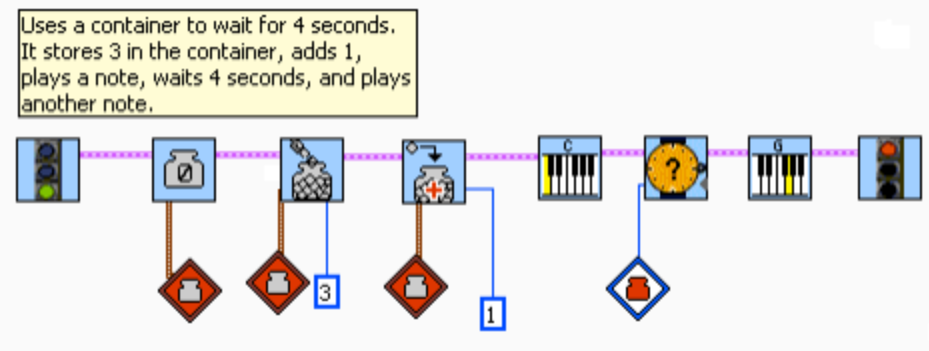


means to wait for 5 seconds, and the following two examples of code work as described:



L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.



Given how it was used in the above two programs, explain the meaning of the following:

Pseudocode

5. **(5 min)** Use **pseudocode** to design a program that does the following:
   a. Makes your robot move forward for a **random amount of time (this means <u>the robot chooses randomly</u>--this value should change every time it is run).** The random amount should be between 1 and 5 seconds. (Note: This is not between 0 and 5, but 1 and 5)
   b. When it stops makes a sound using  . *(Hint, be sure your random time can be between 1 and 5, not 0 and 8.)*
   c. Next, when the touch sensor is pressed, the robot should back up twice as far as it originally traveled forward.
   d. Insert the SET DISPLAY icon from the RCX communication sub-menu into your program for both directions of movement. The amount of time moved should be displayed on the RCX display screen for both forward and backward movements.

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

Next, implement these ideas in RoboLab

In this lab you will create a new Inventor Level 4 program using the touch sensor.

Some helpful Commands

We can use the LCD Screen  to display a number stored in a container which we will want to do in this lab. Because this display is a form of communication, this icon can be found in the communications menu under the Communications  icon.

You will find that the  is **a particularly useful icon in today's lab** because **it needs to be used to put a random number into a container.**

6. **(2 min) Explain why when you program the robot for today's task, you need to use**  **and not this**  **.**

Programming

To create your own RoboLab program, open RoboLab in the Programmer: Inventor 4 mode. Name the program *yourusername1-yourusername*-L5. Implement your ideas on your robot by adding any needed sensors and actuators to your robot, writing this RoboLab program, and trying it on your robot. Also, be sure to include appropriate comments in your code such as each team member's name and role. Comment blocks are required in all labs.

7. **(2 min)** You might want to consider using Wait for a very small number of hundredths of a second icon at one or more points in your program... Discuss why this might be useful when using a touch sensor.

L05: CSC 126, Introduction to Robotics

This is a team assignment designed as an in-class activity.

---

8. **(5 min)** Discuss the most important things you learned in doing the lab today

9. **(5 min)** ***Comments and Suggestions*****: Discuss this lab in your team and write a paragraph that summarizes your team's reaction to this lab. If there are any problems you encountered or any questions that remain, please ask! Also, be sure to include any suggestions you have for how this lab could be improved.**

To Earn Credit

- **Please note:** Your submission in Moodle means that you were fully involved with all of the work and all of the thinking described above. If you submit work in which you were not wholly a collaborative participant, not only will you not have done the learning needed for the exams, it will be considered academic dishonesty and will be treated as such.
- **Submit the pdf of your completed lab activity in Moodle, AND upload the program *yourusername1-yourusername2*-L5.vi**