

# CPSC 2151

## Lab 9

Due: At the end of Lab

In this lab you will be working with a formal reasoning tutorial that uses a verifiable language called resolve. You will need to reason formally about the code and correctly complete a Boolean expression at the end of the code that can be proven to be true to move onto the next activity.

### The Resolve Language

The resolve language that is used is not drastically different from normal programming languages, but there are a few things we want to point out:

**:=** Is the assignment operator in Resolve. In Java we just use the **=** as an assignment operation, but Resolve uses **:=**

**=** is the equivalence operator in Resolve, as opposed to **==** in Java

**Remember** is used to save the current state of all variables in the program. So when a Remember statement is hit in the resolve code, the compiler will store the current state of all variables as the variable name with the **#** in front of it. So if you have variables I, J and K when you hit a Remember statement, it will save their current state as **#I**, **#J**, **#K**. We can refer to those original values in our Boolean statements.

**#var** is Resolve's way of referring to the stored state of the variable var from the last Remember statement

**Confirm** is where we can put a Boolean expression that the Resolve compiler will verify as being either true or false. It's kind of like an assert statement in Java. In these exercises we will complete the Boolean expressions in the Confirm statements about what we expect to always be true at that point in the code, often referring to the saved state of our variables. The Resolve compiler can then automatically prove for us that the expression we entered will always be true for that section of code.

**Read(var)** is how Resolve reads input from the user into our variable var. This makes it more difficult to reason about our code because we have no idea what value the user may have entered. However we can still reason about how the value in var will change as the program executes.

### Instructions

Follow the link on Canvas to get to the Resolve website, and click on the Begin to Reason link to launch the tutorial. The Tutorial starts off with a few simple examples that use actual values of the variables (such as 1, 2, 3). These are just meant to introduce you to the interface.

Tutorial 0 - Variables and Values		BeginToReason	
<div> <a href="#">&lt;&lt; Prev</a> <a href="#">Next &gt;&gt;</a> </div>		<div>Click here for quick video instructions</div>	
<div> <b>Activity:</b>            Please complete the <b>Confirm</b> assertion(s) by entering an expression for <code>/* expression */</code>, then check correctness.         </div>		<pre> 1 Facility BeginToReason; 2   uses Integer_Ext_Theory; 3 4   Operation Main(); 5   Procedure 6     Var I, J, K: Integer; 7 8     I := 2; 9     J := 3; 10 11    K := I; 12    If (J &gt; I) then 13      K := J; 14    end; 15 16    Confirm K = /*expression*/; 17  end Main; 18 end BeginToReason;           </pre>	
<div> <b>Reference Material:</b>  <code>:=</code> is the <i>assignment operator</i> </div>			
<div> <a href="#">&lt;&lt; Prev</a> <a href="#">Next &gt;&gt;</a> </div>		<div>Click here to check correctness</div>	
<a href="#">Reload lesson</a>	<a href="#">End survey</a>	This research is funded in part by NSF grants CCF-1161916 and DUE-1611714. This is a joint effort involving Clemson University, Florida Atlantic University, and Indiana University Southeast.	

The main windows contains our code. At the bottom of the code is a confirm statement we must complete to move on by replacing the `/*expression*/` comment with the value we expect K to be equal to at this point. This example has actual values for our variables, so we can use an actual value as well. The Activity window on the left has our instructions, while the Reference Material window will contain reminders about resolve syntax.

Once you have completes the Confirm statement, click on the green bar on the bottom right to check to see if you are correct. If you are you will move to the next problem.

After the first few tutorial problems you will then start the actual activity which contains 8 lessons and 2 challenge problems. Once you complete those, you will be given a code showing that you completed the assignment. DO NOT CLOSE THE WINDOW YET. Call a TA over who will confirm that you completed the assignment, then give you a paper problem that you must reason about formally. Once you complete the paper assignment you will submit that to the TA and will be able to leave.

## Groups

You may, but are not required to, work with a partner on this lab. Your partner must be in the same lab section as you, not just the same lecture section. If you work with a partner, only one person should submit the assignment. You should put the names of both partners in a comment at the top of the assignment. Remember to actively collaborate and communicate with your partner. Trying to just divide up the work evenly will be problematic.