These instructions are very brief. Ask questions whenever you can't figure out what to do.

The goals of today's lab are to learn how to:
- create Karel programs and their worlds
- use the Karel simulator to execute Karel programs
- write and test solutions to Karel problems

1) The instructor will give a brief overview of how to create (and save) Karel worlds, and how to create, save and execute Karel programs (Use your space on either the M: or N: drives to store your files. If you don't know how to use these drives, ask your instructor.) Any work not finished in lab today is homework to be done during the week. In parts 3, 4 and 5, have your instructor initial the answer sheet and hand it in at the end of lab. For part 6 you will mail me the solution.

2) In class you received an overview and a sheet with a summary of the Karel language. Use your class notes, the summary and the Karel Manual as your programming reference (there is one manual for every two students. Please treat the manuals with care.) The manual contains a complete explanation of Karel and the Karel Language. Remember that the syntax that we are using is slightly different from the syntax of the book. For a complete description of this version of Karel see: http://math.otterbein.edu/Class/Csc100/Karel/web/Karel/karel.htm

3) The "Stair Cleaning Problem"
   a) Read the "stair cleaning" problem in section 3.5 (pp. 29-30) from the Karel the Robot manual. Examine the stair-cleaning program on p 29. Be sure you understand the program, and then type it in using the editor.
   b) Create the World for this problem in the World Builder part of the Karel program.
   c) Run the "stair-cleaning" problem on the Karel Simulator using the Single Step mode. Examine the behavior of the robot as the code proceeds step-by-step using the single-step mode. When you understand the program, run it in the Run mode.
   d) When it is working correctly, show it to the instructor and ask to be checked off on the "stair cleaning" problem.

4) Now do the "fetch-the-morning-newspaper problem" - problem 5 at the back of chapter 2.
   Read the problem; create the world; solve the problem; type in the program; execute it. First use the single step mode and later the regular mode. If there are errors, fix them and re-execute the program until you have a correctly running program. When you have success, show the instructor and ask to be checked off on the "fetch the newspaper" problem on your answer sheet.

5) Next is the "steeplechase problem" - problem 6 at the end of chapter 4.
   a) Read the problem; create the world; solve the problem; type in the program and execute it. Repeat that process until you have a successful program.
   b) Although you can use the world shown in figure 4.5 of the problems in Chapter 4 to test your program, the program must work for any height barriers (of one, two or three blocks) not just the ones shown in the diagram. To make the robot go up over a hurdle, as it's going up (or
down) it must constantly test to see if it has gotten to the top (or bottom) yet. Test your
program on a different world that meets these specifications.

c) For grading, your program will be tested on a world that has different heights for the different
barriers. Your instructor will ask you to change your existing world so he can see this happen.
The test world will assume that Karel is at First Street and First Avenue and facing east. You
**must** follow the specifications exactly. When you have success, show the instructor and ask
to be checked off on the steeplechase problem.

5) Start on the problem below and continue this during the coming week. You will have to do this
work in the programming lab since the Karel simulator is not installed in other labs. During the
week (between 8:15 and 4:30) you can borrow a Karel the Robot book from our department
secretary (leaving your ID card behind to help you remember to return the book!) for use within
this building only.

Solve the "super-duper steeplechase problem" (Figure 5-26 is one example
of such steeplechase.) In the super-duper steeplechase, the hurdles can be
between any two avenues and can be of any height and any width (all
hurdles have at least one avenue separating them - see diagram). On some
block along First Street, there is a beeper that marks the end of the
steeplechase (an example is shown in Figure 5.26) There are no other
beepers. Karel must continue until s/he encounters a beeper.

Send me the solution as an attachment to email. My address is wjt@mail.plymouth.edu The
filename must be your_last_name followed by a ".kp" extension. For example, if I were sending the
file it would be taffe.kp).

- Be sure to use the correct name. If your name is missing, you won’t get credit
- Don’t send a world: I will test the files using my own version of a “world”. You must follow the
  specifications exactly, meaning that your program must work from the position and orientation
  of Karel as specified in the problem. If your program assumes a different specification, it won’t
  work. Your solution must work on any world that meets the specifications of the problem.
- This is due in one week; mail it to me before the start of lab next week.