

Karel Contest

Due: 5:00 P.M., Wednesday, April 19

The first assignment is designed to teach you about Karel's capabilities and to reinforce basic programming concepts. Karel is a good medium for teaching such concepts but also has other virtues. In particular, programming Karel can be a lot of fun. This contest gives you the opportunity to explore this aspect of Karel's world. It is entirely optional, but gives you a chance for some extra credit in the course.

Your mission, should you decide to accept it, is to program Karel to solve an interesting and exciting problem of your own choosing. You can program Karel to produce the definitive work of computer art, to illustrate a story, or to tackle a conceptually difficult task. The entries will be judged by the CS 106J staff (see official rules below), and a prize will be awarded in each of two categories:

- *Aesthetic merit.* This prize is awarded to the program that, in the opinion of the judges, has the greatest literary, artistic, or entertainment value.
- *Algorithmic sophistication.* This prize is awarded to the Karel program that solves the most challenging task in the most interesting way.

Prizes

The first prize in each of the categories will be that we will replace whatever individual score most negatively affects your grade with a 100%. Thus, if you are one of the two Karel contest winners, and you bomb an assignment, the midterm, or even the final, we will overlook that misstep and count it as a 100%. By putting in a little extra effort now, you could reduce substantially the amount of pressure later in the course, which can come in handy. Best of luck!

New extensions

The current implementation of Karel includes a few extensions beyond those described in the text. To use them, you need to import the "**extensions**" library, as follows:

```
import "extensions";
```

1. *Better support for animation.* The old version of Karel made it difficult to write animated programs because there was no good way to control the speed of the display. The new version includes a new built-in function

```
pause (milliseconds) ;
```

that suspends Karel's operation for the specified number of milliseconds. The usual approach to using this statement is to have Karel perform some operation and then pause for a short time (typically on the order of 20 milliseconds or so) to give the display time to catch up.

2. *Painted squares.* The extensions library makes it possible for Karel to paint the corner in any of JavaScript’s colors, which are listed in Figure 2-7 on page 61 of the reader. For example, calling

```
paintCorner("Red");
```

paints the current corner red.

The extensions library also defines the condition `cornerColorIs`, which allows Karel to test whether a corner is painted if a particular color. For example, the `if` statement

```
if (cornerColorIs("Red")) . . .
```

would execute the body of the `if` statement if Karel is standing on a red square.

3. *Random behavior.* The extensions library defines the condition `random()`, which is true 50% of the time. You can also include a number inside the parentheses to specify a particular probability. This probability must be between 0.0 (which means that the event never happens) and 1.0 (which means that it always does).
4. *Logical operators.* The extensions library enables the JavaScript operators `&&` (signifying AND), `||` (signifying OR), and `!` (signifying NOT), as described in the section on “Logical operators” on page 80.

Official rules:

1. Only students registered in CS 106J are eligible to submit contest entries.
2. Only one entry per student will be accepted.
3. All entries must be submitted electronically and must be received by 5:00P.M. on Wednesday, April 19. Late entries will not be accepted.
4. Each submission must consist of a Karel program and one or more worlds for execution. In addition, you may submit a short text explanation, not to exceed 250 words, describing what Karel is doing.
5. Karel programs must limit themselves to the language features described in the text, along with the extensions described earlier in this handout.
6. Contest entries should be sensitive to Stanford’s individual and cultural diversity. Programs or narratives that have the effect of perpetuating negative stereotypes will not be eligible for prizes.
7. Contest entries will be evaluated initially by Eric Roberts, Jerry Cain, and Jason Chen. The best entries will then be evaluated by representatives of the entire CS 106 course staff, which will choose the winners and runners-up in each category.