

Format of a Karel robotics program

```
Program <program-name>
{
    <list of tasks>

    Main()
    {
        <list of instructions>
    }
}
```

The `<program-name>` is a descriptive name, chosen by the programmer, beginning as a single letter that is followed by any number of additional letters, numbers, or hyphens.

Note: all text in Karel is case-sensitive (e.g. `Program` \neq `program`).

A `<list of instructions>` is an ordered set of instructions each of which must be either `<primitive>`, `<block>`, `<control>`, or `<task>`. Tasks are used in a program in the same way as primitives. Task definition is discussed at the end of this handout.

The `<primitive>` instructions are the simple instructions that every Karel robot is designed to perform. There are only five primitive instructions. They are:

```
move();
turnleft();
pickbeeper();
putbeeper();
turnoff();
```

A `<block>` instruction is a collection of instructions taken as a whole unit. They are noted by their appearance inside curly braces:

```
{           <list of instructions>
}
```

A `<control>` instruction alters the normal sequential path of instruction execution. There are four control instructions, two for repetition: `repeat`, and `while`, and two for selection: `if`, and `if/else`. Their forms are:

```
repeat <n> times
       <instruction>
```

where `<n>` is any integer number (for example, 5, 42, 1234).

```
while ( <condition> ) do
       <instruction>
```

Note that a condition must appear inside parentheses, as indicated in the form above. Selection instructions have the following forms:

```
if ( <condition> )
    <instruction>
```

```
if ( <condition> )
    <instruction>
else
    <instruction>
```

A <condition> that the robot can sense results in an answer of yes or no from the appropriate sensor. The result of a condition is used in the control statements. The conditions that the robot can test are:

Direction sensor

```
facing-north
facing-east
facing-west
facing-south
not-facing-north
not-facing-east
not-facing-west
not-facing-south
```

Camera sensor for wall detection

```
front-is-clear
left-is-clear
right-is-clear
front-is-blocked
left-is-blocked
right-is-blocked
```

Microphone sensor for beeper detection

```
next-to-a-beeper
not-next-to-a-beeper
any-beepers-in-beeper-bag
no-beepers-in-beeper-bag
```

A <task> module is a block that is referenced from within one or more other blocks. The <task-name> is a descriptive name, chosen by the programmer, beginning as a single letter that is followed by any number of additional letters, numbers, or hyphens. In a Karel program every task that is to be used in the program must be defined in the program before it is ever used by another part of the program. The basic format for a task is:

```
Task <task-name>()
{
    <list of instructions>
}
```