

## Solutions for Section #1

---

### 1. United Nations Karel

```
"""
File: UnitedNationsKarel.py
-----
The UnitedNationsKarel program builds houses at corners
marked by rubble.
"""

from karel.stanfordkarel import *

def main():
    while front_is_clear():
        if on_beeper():
            pick_beeper()
            backup()
            build_house()
        if front_is_clear():
            move()

def build_house():
    """
    This function builds a beeper house on stilts.
    Pre-condition: Karel faces east at bottom of left stilt
    Post-condition: Karel faces east at bottom of right stilt
    """
    turn_left()
    put_three_beeper()
    move()
    turn_right()
    move()
    turn_right()
    put_three_beeper()
    turn_around()
    move()
    turn_right()
    move()
    turn_right()
    put_three_beeper()
    turn_left()

# CONTINUED ON NEXT PAGE...
```

```
def put_three_beeper():
    """
    Creates a line of three beepers.
    Pre-condition: Karel is in the first square in the line
    Post-condition: Karel is in the last square in the line
    """
    put_beeper()
    move()
    put_beeper()
    move()
    put_beeper()

def backup():
    """
    Backs up one corner, leaving Karel facing in the same
    direction. If there is no space behind Karel, it will run
    into a wall.
    """
    turn_around()
    move()
    turn_around()

def turn_around():
    """
    Rotates Karel 180 degrees.
    """
    turn_left()
    turn_left()

def turn_right():
    """
    Rotates Karel 90 clockwise.
    """
    turn_left()
    turn_left()
    turn_left()

if __name__ == "__main__":
    execute_karel_task(main)
```

Try downloading the PyCharm project for this section from the course website and running this program on your own computer. What would you do if you wanted the houses to be taller? Is there any way you could improve the style of the solutions?

## 2. Karel Defends Democracy

```
"""
File: ChadKarel.py
-----
ChadKarel is a program in which Karel cleans up hanging chads
from a ballot.
"""

from karel.stanfordkarel import *

def main():
    """
    We split the logic into a loop to process columns and advance one
    spot, plus one final call to check the last column.
    """
    while front_is_clear():
        process_column()
        move()
    process_column()

def process_column():
    """
    This function clears chad from the current column, if any.
    Pre-condition: Karel is standing in the center of a column,
    facing east.
    Post-condition: Karel is back in same place/orientation and chad
    has been cleared.
    """
    # If there are chads to clear, clear the chads from the ballot.
    if not on_beeper():
        remove_all_chad()

def remove_all_chad():
    """
    Remove all chads in a given column.
    """
    # Clean the upper corner
    turn_left()
    clean_chad()

    # Clean the lower corner.
    turn_left()
    turn_left()
    clean_chad()

    turn_left() # Face East

# Continued on next page...
```

```
def clean_chad():
    """
    This function clears chad from corner Karel is facing.
    Pre-condition: Karel is facing a corner to be cleared of chad.
    Post-condition: Karel is in the same location/orientation, but
    all chad has been cleared from the corner Karel is facing.
    """
    move()
    while on_beeper():
        pick_beeper()
    backup()

def backup():
    """
    Backs up one corner, leaving Karel facing in the same direction.
    If there is no space behind Karel, it will run into a wall.
    """
    turn_left()
    turn_left()
    move()
    turn_left()
    turn_left()

if __name__ == "__main__":
    execute_karel_task(main)
```

Try downloading the PyCharm project for this section from the course website and running this program on your own computer.

## Style focus for Section 1:

**Comments:** Make sure to comment every function you write by describing what the function does and what the assumptions are before and after it is called. Write your comments so that your program can easily be understood by another person.

**Good function names:** Part of good style is good naming. You want your function name to succinctly describe what it does. Never call a function `do_stuff()`; give it a good specific name like `back_up()`. Be consistent in how you name your functions. In Python, we always use all lowercase letters, with words separated by underscores.

**Short functions:** We could have written our whole program in the `main()` function, but it is not good style and is difficult to follow. Try to break it down into functions that are small, understandable pieces of code and that accomplish one main task. Establish milestones as you work so that you can write some code for one simpler task, test it, and fix it before moving on to the next task in solving a complex problem.

See the CS 106AP Style Guide on the course website (linked to from the sidebar on the homepage) for more important style tips!