Data Structures and Algorithms in Python

Michael T. Goodrich
Department of Computer Science
University of California, Irvine

Roberto Tamassia
Department of Computer Science
Brown University

Michael H. Goldwasser
Department of Mathematics and Computer Science
Saint Louis University

Instructor's Solutions Manual

WILEY

Chapter 2 Object-Oriented Programming

Hints and Solutions

Reinforcement

R-2.1) Hint Think of applications that could cause a death if a computer failed.

R-2.1) Solution Air traffic control software, computer integrated surgery applications, and flight navigation systems.

R-2.2) Hint Consider an application that is expected to change over time, because of changing economics, politics, or technology.

R-2.3) Hint Consider the File or Window menus.

R-2.4) Hint Consider using get and set methods for accessing and modifying the values.

R-2.5) Hint Read about exception handling in Chapter 1.

R-2.6) Hint Read about exception handling in Chapter 1.

R-2.7) Hint Read about default parameter values in Chapter 1.

R-2.8) Hint Try to make the last card over its limit.

R-2.8) Solution

```
for val in range(1,59):
  wallet[0].charge(val)
  wallet[1].charge(2*val)
  wallet[2].charge(3*val)
```

This change will cause the last credit card to go over its limit.

R-2.9) Hint The code should look very similar to `__add__`.

R-2.10) Hint Create a vector of the appropriate length and then set its coordinates.

R-2.10) Solution

```
  def __neg__(self):
    result = Vector(len(self))
    for j in range(len(self)):
      result[j] = −self[j]
```

R-2.11) Hint You will need to define the __radd__ method.

R-2.11) Solution

```
def __radd__(self, other):
  return self + other      # rely on definition of __add__
```

R-2.12) Hint Create a vector of the appropriate length and then set its coordinates.

R-2.13) Hint You should be able to reuse your implementation of __mul__.

R-2.14) Hint Remember that you are returning a single number (not a vector).

R-2.15) Hint Use the isinstance function to determine the operand type.

R-2.16) Hint If we were to increase the stop value, one at a time, at what point would a new value appear in the range?

R-2.17) Hint Review the definition of inheritance diagram, and begin your drawing with object as the highest box.

R-2.18) Hint Your program should output 42, which Douglas Adams considers to be the answer to the ultimate question of life, universe, and everything.

R-2.19) Hint Try it out.

R-2.19) Solution 2^{56} calls to next end on the value 2^{63}, so $2^{56} - 1$ calls can be made before that time.

R-2.20) Hint Think about what happens when a new instance of class Z is created and when methods of class Z are called.

R-2.20) Solution There are two immediate inefficiencies: (1) the chaining of constructors implies a potentially long set of method calls any time an instance of a deep class, Z, is created, and (2) the dynamic dispatch algorithm for determining which version of a certain method to use could end up looking through a large number of classes before it finds the right one to use.

R-2.21) Hint Think about code reuse.

R-2.21) Solution Whenever a large number of classes all extend from a single class, it is likely that you are missing out on potential code reuse from similar methods in different classes. There is likely some factoring of methods into common classes that could be done in this case, which would save programmer time and maintenance time, by eliminating duplicated code.

R-2.22) Hint Be especially careful when the two sequences do not have the same length.

R-2.23) Hint Be especially careful when one sequence is a prefix of another.

R-2.23) Solution

```
def __lt__(self, other):
  for k in range(min(len(self),len(other))):
    if self[k] < other[k]:
      return True
    elif self[k] > other[k]:
      return False
    # otherwise elements are equal thus far...
  # if reached the end, require that self be a strict prefix of other
  return len(self) < len(other)
```

Creativity

C-2.24) Hint Create a separate class for each major behavior.

C-2.25) Hint Use the isinstance function to determine the operand type.

C-2.26) Hint Think about how the internal counter should be initialized.

C-2.27) Hint Consider the difference between the target value and the start of the range, and the step size for that range.

C-2.28) Hint The key is being able to accurately track how many times charge has been called thus far during a month.

C-2.28) Solution

```
def __init__(self, customer, bank, acnt, limit, apr):
  super().__init__(customer, bank, acnt, limit)
  self._apr = apr
  self._charge_count = 0              # new instance variable

def process_month(self):
  self._charge_count = 0              # reset
  ...

def charge(self, price):
self._charge_count += 1
if self._charge_count > 10:
  self._balance += 1                  # assess a $1 fee
...
```

C-2.29) Hint You will need to keep track of how much payment has been received in the current month.

C-2.30) Hint Make sure to test your modified code.

C-2.31) Hint Model your solution after our other subclasses of Progression.

C-2.31) Solution

```
class AbsoluteProgression(Progression):
  def __init__(self, first=2, second=200):
    super().__init__(first)
    self._prev = first−second      # thus second = abs(first-prev)

  def _advance(self):
    next = abs(self.current−self._prev)
    self._prev = self._current
    self._current = next
```

C-2.32) Hint Use the sqrt function in the math module.

Projects

P-2.33) Hint If you have not had calculus, you can look up the formula for the first derivative of a polynomial on the Internet.

P-2.34) Hint You don't have to use GUI constructs; simple text output is sufficient, say, using X's to indicate the values to print for each bar (and printing them sideways).

P-2.35) Hint Use three different classes, for each of the actors, and provide methods that perform their various tasks, as well as a simulator engine that performs the periodic operations.

P-2.36) Hint When a fish dies, set its associated cell back to **None**.

P-2.37) Hint Use random number generation for the strength field.

P-2.38) Hint Create a separate class for each major behavior. Find the available books on the Internet, but be sure they have expired copyrights.

P-2.39) Hint Look up the formulas for area and perimeter on the Internet.