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

Study Guide: Hints to Exercises

WILEY

Chapter 2 Object-Oriented Programming

Hints

Reinforcement

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

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

R-2.3) Consider the File or Window menus.

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

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

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

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

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

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

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

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

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

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

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

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

R-2.16) 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) Review the definition of inheritance diagram, and begin your drawing with object as the highest box.

R-2.18) 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) Try it out.

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

R-2.21) Think about code reuse.

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

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

Creativity

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

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

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

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

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

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

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

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

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

Projects

P-2.33) 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) 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) 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) When a fish dies, set its associated cell back to **None**.

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

P-2.38) 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) Look up the formulas for area and perimeter on the Internet.