

Review of Object-Oriented Programming

Answer the following questions, or write programs as instructed.

1. Define the following terms as they relate to object-oriented programming:
 - encapsulation
 - abstraction
 - inheritance
 - polymorphism
2. Create a class, `Computer`, which has the following attributes: `manufacturer`, `model`, `cpu_speed`, `ram_size`, `storage_capacity`, and `price`. Create two derived classes, `Desktop` and `Laptop`. `Desktop` should have the additional attributes `length`, `width` and `height`, while `Laptop` should have the additional attributes `screen_size` and `weight`. Write methods to display information in a nicely-formatted table. Create an instance of each derived class and display its information.
3. Create a class diagram for `Computer`, `Desktop` and `Laptop` as defined in Q1. Be sure to indicate the inheritance relationship.
4. Create a class, `Prism`, which has the following attributes: `length`, `width` and `height`. Create three derived classes, `RectangularPrism`, `Cube` and `RightTriangularPrism` which inherit the attributes from `Prism`. For each derived class, create the methods `volume()` and `surface_area()` that perform these calculations. In your main program, create a list containing ten randomly-generated shapes. For each shape, display its dimensions, its volume, and its surface area.
5. In mathematics, a **vector** is a singly-directed line segment, or an arrow. Algebraically, a vector can be represented using a point, $P(x, y, z)$, which marks the location of its head. Write a class, `Vector`, which has three attributes: `x`, `y` and `z`. Override the following built-in methods so that the `+`, `-` and `*` operators perform the following actions:
 - `__add__()`: $U(x_1, y_1, z_1) + V(x_2, y_2, z_2) = W(x_1+x_2, y_1+y_2, z_1+z_2)$
 - `__sub__()`: $U(x_1, y_1, z_1) - V(x_2, y_2, z_2) = W(x_1-x_2, y_1-y_2, z_1-z_2)$
 - `__mul__()` and `__rmul__()`: $k * V(x, y, z) = W(kx, ky, kz)$ and $V(x, y, z) * k = W(kx, ky, kz)$Override the `__str__()` function so that a `Vector` displays as `(x, y, z)`. Create instances of two or more vectors, and test your methods.