Department of Computer Science and Engineering
PES University, Bangalore, India

Lecture Notes
Python for Computational Problem Solving
UE23CS151A

Lecture #103

Problem Solving on classes, objects, Polymorphism and Inheritance

By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU

Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Centers, Former Chairperson, CSE, PES University)
Prof. Chitra G M(Asst. Prof, Dept. of CSE, PCPS Anchor – 2022)

Practice Programs

Try solving these problem statements. Solutions for a few are available in this link: https://drive.google.com/file/d/1DbYfPusibwR-W-Nhi2bcGbzQIb0VrLY_/view?usp=drive_link

1. Implement the Shape Hierarchy by creating a "Shape" type as the parent type. Add new types ' Rectangle', 'Circle' and 'Triangle' inheriting from 'shape'. Take the **radius of the circle**, length and width of rectangle, base, height, side1, side2 and side3 for a triangle from the user. Add methods in the sub types to calculate area and perimeter. Create instances of the respective types and test all these calculations.

2. Create a base class 'Vehicle' with attributes - price, model and year. Derive named classes such as Car and Motorcycle from Vehicle. Each subclass should have additional attributes specific to Car and Motorcycle respectively. Create two instances of Car and two instances of Motorcycle and print the instances directly by calling a print function.

3. Create a Bank_Account type. Add balance as its instance variable. Add separate instance methods to deposit the amount and withdraw the amount from the account separately. Add appropriate functionality in withdraw method to check for minimum balance before withdrawing the amount and display proper message. Create two instances of Bank_Account. Initial balance is always 0 for every instance. Test all these functionality in the driver/test code.

4. Create a new type called Home with attributes - num_rooms and num_stories. Create an object of Home by passing these values. Print the object of Home type.
This must print "The house has ___ rooms and ____ stories"

5. Create a type called Complex with real and imaginary parts. Create two objects of this type. Perform addition of these two using +, subtraction using -, multiplication using * and true division using /. Write a complete code to fit into the given driver/test code below.

```
c1 = Complex(1, 4)
c2 = Complex(6, 1)
print(c1)
print(c2)
print("---------------------")
print(c1 + c2)
print(c1 - c2)
print(c1 * c2)
print(c1 / c2)
```

6. Create an Employee type with details like – id, name, age. Create a few instances of Employee. Print the number of objects created. Delete any two objects and again print the count of objects.

7. Create a class for books with attributes like title, author, and genre. Implement a simple library catalog system to add and display book information.

8. Develop a system with classes for employees and departments. Include functionalities like displaying employee details and assigning employees to departments.

9. Create a class for temperature conversion with methods to convert Celsius to Fahrenheit and vice versa. Test the class with sample temperature values.

10. Design classes for basic animals (cat, dog) with attributes like name and sound. Simulate their behavior by displaying the name and the sound they make.

For interested students only:

11. Implement a simple ToDo list application with classes for tasks. Include functionalities like adding tasks, marking them as complete, and displaying the list.

12. Create a library management system with classes for books, patrons, and transactions. Implement functions to handle book checkout, return, and overdue fines.

13. Build a multimedia player application with classes representing different types of media (audio, video, images). Implement features like play, pause, stop, and volume control.

14. Develop a reservation system for a hotel with classes for rooms, guests, and reservations. Implement methods to check room availability, make reservations, and calculate total costs.

15. Design a simple online shopping system with classes for products, customers, and shopping carts. Include features like adding items to the cart, checkout, and order history.

16. Build a music player program where 'Audiofile' is the base class. Implement sub classes 'MP3File' and 'WAVFile' inheriting from 'AudioFile'. Both classes should have a 'play' method but provide different functionalities for playing MP3 and WAV files.

17. Create a base class 'Vehicle' with a method 'calculate_speed'. Derive subclasses 'Car' and 'Bike' from 'Vehicle'. Override the 'calculate_speed' method in each subclass to calculate their specific speed based on different parameters

-END-