



Python OOPs Assignment

support@intellipaat.com

+91-7022374614

US: 1-800-216-8930 (Toll-Free)

Assignment Questions

Q1. Write a Python program to demonstrate multiple inheritance.

1. **Employee** class has 3 data members **EmployeeID**, **Gender** (String), **Salary** and **PerformanceRating**(Out of 5) of type int. It has a get() function to get these details from the user.
2. **JoiningDetail** class has a data member **DateOfJoining** of type **Date** and a function **getDoJ** to get the Date of joining of employees.
3. **Information** Class uses the marks from **Employee** class and the **DateOfJoining** date from the **JoiningDetail** class to calculate the top 3 Employees based on their Ratings and then Display, using **readData**, all the details on these employees in Ascending order of their Date Of Joining.

Q.2 Write a Python program to demonstrate Polymorphism.

1. Class **Vehicle** with a parameterized function **Fare**, that takes input value as fare and returns it to calling Objects.
2. Create five separate variables **Bus**, **Car**, **Train**, **Truck** and **Ship** that call the **Fare** function.
3. Use a third variable **TotalFare** to store the sum of fare for each Vehicle Type.
4. Print the **TotalFare**.

Q3. Consider an ongoing test cricket series. Following are the names of the players and their scores in the test1 and 2.

Test Match 1 :

Dhoni : 56 , Balaji : 94

Test Match 2 :

Balaji : 80 , Dravid : 105

Calculate the highest number of runs scored by an individual cricketer in both of the matches. Create a python function **Max_Score (M)** that reads a dictionary **M** that recognizes the player with the highest total score. This function will return (Top player , Total Score) . You can consider the Top player as String who is the highest scorer and Top score as Integer .

Input : **Max_Score**({'test1':{'Dhoni':56, 'Balaji': 85}, 'test2':{'Dhoni': 87, 'Balaji':200}})

Output : ('Balaji ' , 200)

Q4. Create a simple Card game in which there are 8 cards which are randomly chosen from a deck. The first card is shown face up. The game asks the player to predict whether the next card in the selection will have a higher or lower value than the currently showing card.

For example, say the card that's shown is a 3. The player chooses "higher," and the next card is shown. If that card has a higher value, the player is correct. In this example, if the player had chosen "lower," they would have been incorrect. If the player guesses correctly, they get 20 points. If they choose incorrectly, they lose 15 points. If the next card to be turned over has the same value as the previous card, the player is incorrect.

Q5. Create an empty dictionary called `Car_0`. Then fill the dictionary with Keys : color , speed , X_position and Y_position.

```
car_0 = {'x_position': 10, 'y_position': 72, 'speed': 'medium'} .
```

- a) If the speed is slow the coordinates of the X_pos get incremented by 2.
- b) If the speed is Medium the coordinates of the X_pos gets incremented by 9
- c) Now if the speed is Fast the coordinates of the X_pos gets incremented by 22.

Print the modified dictionary.

Q6. Show a basic implementation of abstraction in python using the abstract classes.

1. Create an abstract class in python.
2. Implement abstraction with the other classes and base class as abstract class.

Q7. Create a program in python to demonstrate Polymorphism.

1. Make use of private and protected members using python name mangling techniques.

Q8. Given a list of 50 natural numbers from 1-50. Create a function that will take every element from the list and return the square of each element. Use the python map and filter methods to implement the function on the given list.

Q9. Create a class, Triangle. Its init() method should take self, angle1, angle2, and angle3 as arguments.

Q10. Create a class variable named `number_of_sides` and set it equal to 3.

Q11. Create a method named `check_angles`. The sum of a triangle's three angles should return True if the sum is equal to 180, and False otherwise. The method should print whether the angles belong to a triangle or not.

11.1 Write methods to verify if the triangle is an acute triangle or obtuse triangle.

11.2 Create an instance of the triangle class and call all the defined methods.

11.3 Create three child classes of triangle class - isosceles_triangle, right_triangle and equilateral_triangle.

11.4 Define methods which check for their properties.

Q12. Create a class isosceles_right_triangle which inherits from isosceles_triangle and right_triangle.

12.1 Define methods which check for their properties.