**Python Practical Test: OOP + Core Concepts (Beginner to Intermediate) - EVENING TEST**

**Duration:** 3 Hours  
 **Total Questions:** 25  
 **Instructions:**

* Answer all questions.
* Code must be syntactically correct and executable.
* Where applicable, include both class definitions and instantiation examples.

**SECTION A: Core OOP Concepts (Classes, Objects) – 10 Marks**

1. **(2 marks)** Define a class Car with attributes brand, model, and year. Include a method get\_description() that returns a formatted string describing the car.
2. **(1 mark)** Create an instance of the Car class for a "Toyota Corolla" manufactured in 2020.
3. **(2 marks)** Add a class variable to Car to keep track of the total number of cars created. Display the count after creating three instances.
4. **(2 marks)** Modify the Car class to include a list of features. Add a method has\_feature(feature) that returns True if the feature exists.
5. **(3 marks)** Create a class Garage with a list of Car objects and methods to:  
     
   * add\_car(car) – adds a car to the garage
   * get\_cars\_by\_brand(brand) – returns all cars of a given brand

**SECTION B: Inheritance – 10 Marks**

1. **(2 marks)** Create a base class Animal with attributes name and species. Add a method make\_sound() that prints a generic animal sound.
2. **(2 marks)** Create a subclass Dog that inherits from Animal. Add an attribute breed and override make\_sound() to print "Woof!".
3. **(2 marks)** Create a subclass Cat with an additional attribute color and a method purr() that prints "Purr...".
4. **(2 marks)** Create a list of Animal objects (some Dogs, some Cats). Use a loop to call make\_sound() for each.
5. **(2 marks)** Write a function count\_species(animals) that returns a dictionary with species as keys and counts as values.

**SECTION C: Data Structures inside Classes – 5 Marks**

1. **(2 marks)** Create a class Recipe with a dictionary to store ingredient: quantity. Add methods to:  
     
   * add\_ingredient(name, qty)
   * total\_ingredients()
2. **(1 mark)** Create a class Cookbook with a list of Recipe objects. Add a method to list all ingredients across recipes.
3. **(1 mark)** In Cookbook, maintain a set of all unique recipe names.
4. **(1 mark)** Add a tuple to Recipe to store the created\_date in the form (day, month, year).

**SECTION D: Functions, Loops, and Logical Thinking – 10 Marks**

1. **(2 marks)** Create a class Account with methods:  
     
   * deposit(amount)
   * withdraw(amount)
   * get\_balance()
2. **(1 mark)** Simulate 4 deposits and 2 withdrawals using a loop and print the final balance.
3. **(2 marks)** Create a class Inventory with a dictionary items (item\_name: quantity). Include methods to:  
     
   * add\_item(name, qty)
   * remove\_item(name)
   * get\_total\_items()
4. **(1 mark)** Add a method to Inventory to print all items and their quantities using a loop.
5. **(1 mark)** In a class Club, create a method register\_members(\*names) using variable-length arguments to add member names to a list.
6. **(1 mark)** In Club, write a method that returns the shortest member name using a loop.
7. **(1 mark)** Add a method using **list comprehension** to return all member names that start with the letter 'A'.
8. **(1 mark)** In Inventory, write a method that returns a list of items with quantity greater than 5.
9. **(1 mark)** Write a method that checks if a given item exists in the inventory using the in operator.
10. **(1 mark)** In Recipe, add a method that returns a dictionary of ingredients where quantity > 2.
11. **(1 mark)** In any class, write a method with a **while loop** that counts down from 5 to 1, printing each number.