**Python OOPs Assignment**

**Instructions**

* Write clean and well-commented Python code.
* Test every class by creating objects.
* Submit a single .py file with all answers.

**Section A – Class & Object**

**Q1.** Create a class Student with attributes name, roll\_no, and marks.  
Add a method display\_details() to print student details. Create at least two student objects.

**Q2.** Create a class Car with attributes brand, model, and year.  
Add methods: start() and stop(). Create objects and demonstrate method calling.

**Section B – Constructor**

**Q3.** Create a class Book with attributes title, author, price.

* Initialize them using a constructor.
* Add a method apply\_discount(discount\_percentage) that reduces the price.

**Q4.** Create a class Employee with a constructor that accepts name, id, and salary.  
Add a method show\_info(). Create at least two employee objects.

**Section C – Encapsulation**

**Q5.** Create a class BankAccount with:

* Private attributes: \_\_account\_number, \_\_balance.
* Methods: deposit(amount), withdraw(amount), and get\_balance().

**Q6.** Create a class Laptop with private attributes \_\_brand, \_\_price.  
Provide getter and setter methods for both attributes. Demonstrate how encapsulation works.

**Section D – Inheritance**

**Q7.** Create a base class Person with attributes name and age.  
Create a derived class Employee with an additional attribute salary.  
Add a method display\_employee\_details() and demonstrate inheritance.

**Q8.** Create a base class Vehicle with method move().  
Create derived classes Car and Bike that inherit from Vehicle and override the move() method.

**Section E – Polymorphism**

**Q9.** Create two classes Dog and Cat with a method sound().  
Demonstrate polymorphism by calling sound() using a common interface.

**Q10.** Create a class Calculator with a method add().

* If given two numbers → return their sum.
* If given three numbers → return their sum. (use default arguments)

**Q11.** Create a function animal\_sound(animal) that accepts any object with a sound() method.  
Demonstrate polymorphism by passing different animal objects.

**Section F – Abstraction**

**Q12.** Create an abstract class Shape with an abstract method area().  
Implement it in Circle and Rectangle.

**Q13.** Create an abstract class Employee with abstract methods calculate\_salary() and get\_role().  
Implement Manager and Developer classes.

**Section G – Mini Project**

**Q14. Library Management System**

* Book → attributes: title, author, ISBN.
* Member → attributes: name, member\_id.
* Library → methods:
  + add\_book(book)
  + remove\_book(isbn)
  + issue\_book(isbn, member)
  + return\_book(isbn, member)

**Q15. Online Shopping Cart**

* Product → attributes: name, price.
* Cart → methods: add\_product(product), remove\_product(product), calculate\_total().
* Demonstrate with multiple products.