

## **PRACTICE PROBLEMS**

1. Create a class 'Rectangle' with data Members: length, breadth  
Methods:  
    area() - Prints the area.  
    perimeter() - Prints the perimeter.  
Create a class 'Square' with data member 'side' which inherits 'Rectangle' and prints its area & perimeter.
2. Create a class named 'Worker' having the following:  
Data Members: Name, Age, Phone number, Address, Job Level, Salary.  
Methods:  
    printSalary() - Prints salary of the members.  
    showDetails() - Shows the details of members.  
Create two classes 'Employee' and 'Manager' which inherits the 'Worker' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively.  
Salary of Employee:  $20000 + \text{Job Level} * 25000$   
Salary of Manager:  $30000 + \text{Job Level} * 25000$   
Print the details and salary of 'Employee' and 'Manager'.
3. Create a class 'School' with data members: sch\_name, est\_year having 3 constructors:  
    School() - In this constructor default value for sch\_name: 'DEFAULT' and est\_year: '2000'.  
    School(sch\_name) - sets the value for school name  
    School(sch\_name, est\_year) - sets the value for school name and established year.
4. Create a class 'Address' with data members: streetnum, city, state, country and constructor Address(streetnum, city, state, country). Create a 'Student' class with data members: rollno, stu\_name, address of type 'Address'.  
Print the rollno, stu\_name and address of a student.
5. Create class 'Shape' with data members: length1, length2  
Method: calculate\_area() - To calculate area.  
Create class 'Square' with data member: side and override method calculate\_area(). Create class 'Triangle' and 'Circle' with data member side and radius respectively. Override method calculate\_area().

6. Write a program for a small calculator which performs 4 operations (+, -, \*, /). By using polymorphism.
7. Create a class called Book which contains data members Book title,(Instance of author class) Author, Price of book. Create another class Author which contains data members name, email id, phone number, gender. Write a function for adding book details.
8. For the above class Write another function to delete book entry according to book title.
9. For the above class Display all the female authors.
10. Write a function that would display the contact details of the author when the book title is given as input.
11. Write a menu driven program in java to check whether a given number is a prime number or palindrome number. Create classes of your choice and have a method to check whether a given number is a prime number or palindrome number.
12. Create an abstract class Bank, which is having member function getRateOfInterest() which will return the interest rate. Write 2 different classes SBI (interest rate 12%) and PNB(interest rate 13.4%) which extends the class Bank. Write a main class TestBank which returns the interest rate for the input bank .
13. Multiple inheritance (extending more than one class) is not supported through class in java. Write a java program that can implement the idea of multiple inheritance. (Hint: use interface)
14. Imagine a publishing company that makes both book and audiocassette version of its works. Create a class publication that stores the title(a string ) and price(type float) of a publication. From this class derive two classes; book, which adds a page count(type int) and tape, which adds a playing time in minute (type float). Each of these three classes should have getData() function to get its data from the user at the keyboard, and a putData() function to display its data.  
Write main() program to test the book and tape classes by creating instances of them asking the user to fill in data with getData(), and then displaying the data with puData().
15. Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent toll. Mostly they do, but sometimes a car goes without

paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called tollBooth. The two data are a type unsigned int hold the total number of cars and a type double to hold the total amount of money collected. A constructor initialize both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to cash total. Another function called nonpayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals.

Include a program to test this class. This program should allow the use5 to push one key to count a paying car, and another to count a nonpaying car. Before exit, it should print both total cars and total cash.