

1.)

Draw class diagram and develop logic as per the following specifications: The class City has name, area, state, population as attributes. The attributes area (in sq. m) and population are of type long. Code the constructor and toString() method. The main() method must store data of cities in a collection and is menu driven with the following options: 1. Add new city, 2. Sort by population, 3. Sort by area. (Hint: Use Comparator Interface)

2.)

Develop an abstract class Employee with ID, name and address as arguments and an abstract method computeSalary():double. The classes ContractEmployee and RegularEmployee inherit Person. ContractEmployee has hoursWorked and hourlyWage as its attributes and the RegularEmployee has basic and allowance as attributes. The rules for the computation of salary are as follows: 1. For Contract employees the salary is product of hoursWorked and hourlyWage. 2. For Regular employee the salary is basic + HRA(20 % of basic) + allowance. Write the Demo class with main() method to create objects of both ContractEmployee and RegularEmployee and compute the salary. Draw the class diagram.

3.)

Draw the class diagram and develop a class Voter (with ID, name, gender, age as private instance members), define the parameterized constructor and format the output as: ID: 1234, Name: ABC, Gender: Male, Age:45. The task is to store data of voters in an ArrayList. The main () method must be menu driven with the following options: 1. Add New Voter, 2. Print all Voter data, 3. Sort based on age.

4.)

Create a class Book with BName, BId, BAuthor, and YOP (Year of Publication). Use proper getter and setter methods. BName must not have any special symbols except '-', BId must not have any whitespace and special symbols, BAuthor must not have any special characters and digits, and YOP should contain only a 4-digit number. Use toString () to format the details of the book. And enhance the setter methods to throw an exception if the data passed to setter is not a valid data. Create 2 objects in main () and access the methods using these objects. Display the details

5.)

1. Create a Class Engine with attributes engineID (int), engineType (String), horsepower(int), engineWeight (float). add constructors, getter, setters and toString () for Engine class. Enhance the setters in such a way that if invalid data is sent the setter will generate an appropriate exception. Create 2 objects in main () and access the methods using these objects. Display the details.

6.) Explain Final Class, Final method, Final Variable with Example