Name of the Experiment: The objects of Dog class can walk and ~~barks~~ bark. The proof procedure of each action is known and its business logic consists of displaying the way, But there's another unknown functionality of eating which depends on the type of dogs like American Dog, Chinese dog, Australian dog, Canadian dog, Indian dog, and Hungarian dog. The business logic of this action will be implemented showing the procedure of eating of each type of dog. Write a Java program to demonstrate the explained problem.

Introduction: We have to define a base ~~interface~~ abstract class named Dog. Then we will have to define 6 derived class as per mentioned in the problem statement. And finally we will define a Main class.

Objective:-

- to learn how abstract classes work
- to learn how ~~we~~ abstract class can be used to solve problems.

Analysis: After analysing our problem we have found following components of our problem

- we have to define a base abstract class Dog which will have an abstract method named eat.
- we will have to define 6 derived classes which will implement the abstract eat method
- we will have to also define a Main class which will contain the main method.

From above analysis the conceptual class diagram is given below in figure - 1

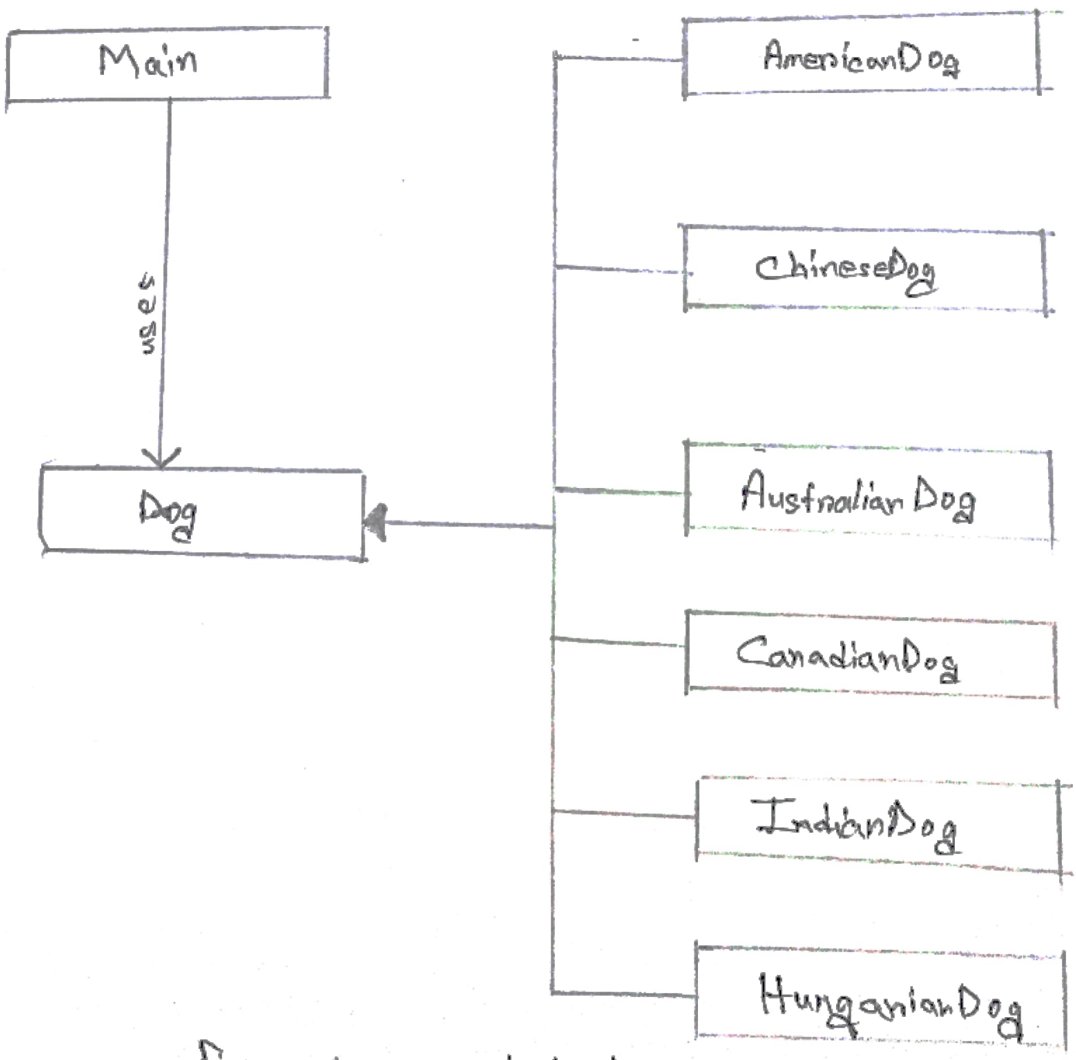

Figure-1: conceptual class diagram

Design : from above analysis our class design descriptions are given below.

- class Dog : an abstract class containing an abstract function
  - ☑ Data Members :
  - ☐ Methods :
    - walk : prints a message that dog is walking
    - sleep : prints a message that dog is sleeping
    - bark : prints a message that dog is barking
    - eat : an abstract method
- class AmericanDog :
  - ☐ Methods :
    - eat : overridden eat method
- class chineseDog :
  - ☐ Methods :
    - eat : overriden eat method
- class AustralianDog :
  - ☐ Methods :
    - eat : overriden eat method
- class CanadianDog :
  - ☐ Methods :
    - eat : overriden eat method
- class IndianDog :
  - ☐ Methods :
    - eat : overriden eat method
- class HungarianDog :
  - ☐ Methods :
    - eat : overriden eat method

From above design analysis the architectural class design is given below,

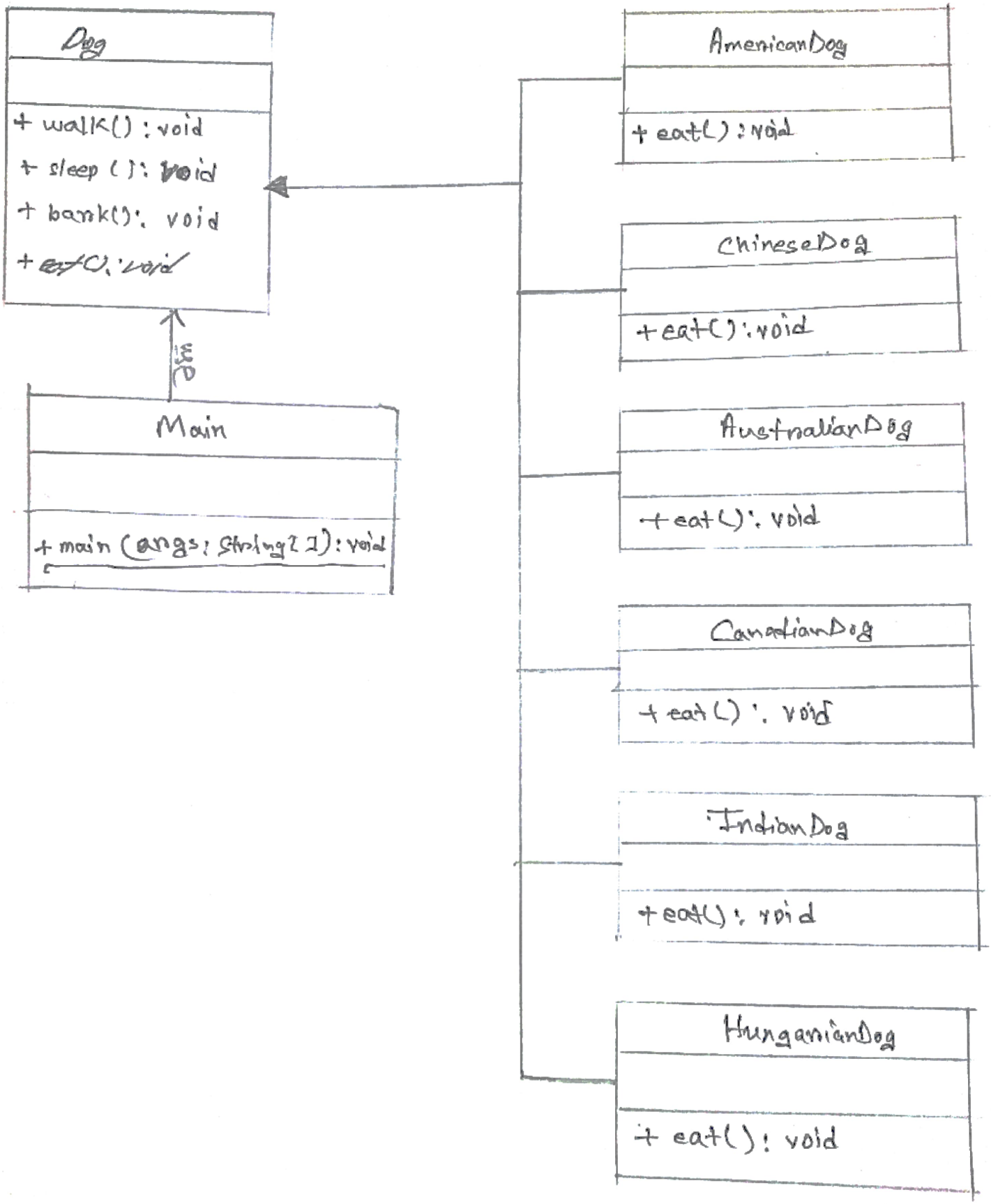

Figure-2: architectural class diagram

from above design description, the pseudocods of the methods are given below:-

```
walk() :
    print walking

sleep() :
    print sleeping

bark() :
    print barking

AmericanDog :: eat() :
    print American Dog eating

ChineseDog :: eat() :
    print Chinese Dog eating

AustralianDog :: eat() :
    print Australian Dog eating

CanadianDog :: eat() :
    print canadian Dog eating

IndianDog :: eat() :
    print Indian Dog eating

HungarianDog :: eat() :
    print Hungarian Dog eating

main(args) :
    create Dog object
    Assign different derived Dog class objects
    and use them to demonstrate utility.
```

Implementation:

* Implementation is attached with the lab report *

Conclusion:- We have create a base abstract Dog class and then defined a breeds of derived Dog type. And then defined a Main class with main method.