Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Experiment No. 7
Implement a program on single inheritance.
Date of Performance:
Date of Submission:

Aim: To implement the concept of single inheritance.

Objective: Ability to design a base and child class relationship to increase reusability.

Theory:

Single inheritance can be defined as a derived class to inherit the basic methods (data members and variables) and behaviour from a superclass. It's a basic is-a relationship concept exists here. Basically, java only uses a single inheritance as a subclass cannot extend more superclass.

Inheritance is the basic properties of object-oriented programming. Inheritance tends to make use of the properties of a class object into another object. Java uses inheritance for the purpose of code-reusability to reduce time by then enhancing reliability and to achieve run time polymorphism. As the codes are reused it makes less development cost and maintenance. Java has different types of inheritance namely single inheritance, multilevel, multiple, hybrid. In this article, we shall go through on basic understanding of single inheritance concept briefly in java with a programming example. Here we shall have a complete implementation in java.

Syntax:

The general syntax for this is given below. The inheritance concepts use the keyword 'extend' to inherit a specific class. Here you will learn how to make use of extending keyword to derive a class. An extend keyword is declared after the class name followed by another class name. Syntax is,

```
class base class
{…. methods
}
class derived class name extends base class
{
methods … along with this additional feature
}
```

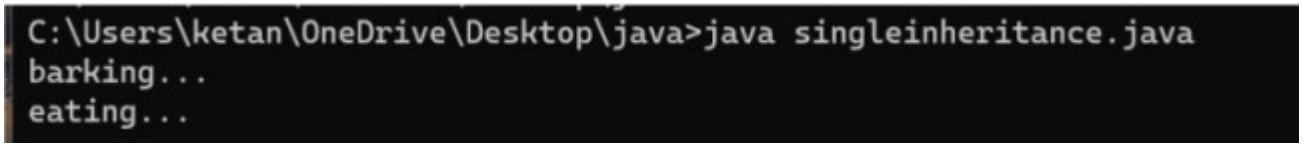
Java uses a keyword 'extends' to make a new class that is derived from the existing class. The inherited class is termed as a base class or superclass, and the newly created class is called derived or subclass.

The class which gives data members and methods known as the base class and the class which takes the methods is known as child class.

Code:

```
1. class TestInheritance {
 public static void main(String args[]) {
 Dog d=new Dog();
d.bark();
 d.eat();
 }
 }
 class Animal {
 void eat() {
 System.out.println("eating...");
 }
 }
 class Dog extends Animal {
 void bark() {
 System.out.println("barking...");
 }
 }
```

OUTPUT:



Conclusion:

Comment on the Single inheritance.

I've encountered the concept of single inheritance in object-oriented programming, and here's my understanding:

Single Inheritance:
- Single inheritance is a fundamental feature in object-oriented programming languages like Java and C++. It allows a class to inherit properties and behaviors from a single parent or base class.
- In single inheritance, a derived class can have only one immediate superclass. This means that the class hierarchy forms a linear or one-dimensional structure.
- Single inheritance promotes simplicity and reduces the chances of ambiguity in method resolution, making it easier to understand and maintain the code.

While single inheritance has its advantages in terms of code organization and preventing certain complexities, it may not capture the full spectrum of relationships in real-world scenarios. In such cases, multiple inheritance, which allows a class to inherit from multiple base classes, might be a more suitable approach. However, single inheritance remains a commonly used and straightforward method of class hierarchy in many programming languages.