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:

```
class SingleInheritancetest{
    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:

```
C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>javac SingleInheritancetest.java

C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>java SingleInheritancetest.java
barking...
eating...
```

Conclusion:

Comment on the Single inheritance.

- Single inheritance is a type of inheritance in which a subclass inherits from a single superclass. It is one of the most common and simple forms of inheritance in object-oriented programming. Single inheritance allows us to reuse the methods and fields of the superclass in the subclass, and also to add new features or override existing ones in the subclass. Single inheritance represents an IS-A relationship between the subclass and the superclass, meaning that the subclass is a specific type of the superclass.
- For example, in Java, we can use the keyword `extends` to create a subclass that inherits from a superclass. Suppose we have a class called Animal that has some common attributes and behaviors of animals, such as name, age, eat, sleep, etc.