Experiment No. 8
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 behavior 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 a 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 is known as the base class and the class which takes the methods is known as child class.

Code:

```
// Parent class
class Animal
{
   void eat()
    {
      System.out.println("This animal eats food.");
    }
}

// Child class that inherits from Animal
class Dog extends Animal
{
   void bark()
    {
      System.out.println("The dog barks.");
   }
}

// Main class to run the program
```

```
public class Main
{
   public static void main(String[] args)
    {
      // Create an object of the Dog class
      Dog myDog = new Dog();

      // Call methods from both the Dog and Animal classes
      myDog.eat();  // Inherited method
      myDog.bark(); // Dog's own method
    }
}
```

Conclusion:

Single inheritance plays a vital role in Java and other object-oriented languages, offering a structured and efficient way to define relationships between classes. By allowing a child class to inherit properties and methods from a single parent class, it enhances code reusability, maintainability, and clarity.