1. **Write a JAVA program give example for “super” keyword.**

```
class Animal
{
   String name;
   Animal(String name)
   {
      this.name = name;
      System.out.println("Animal constructor called");
   }
   void sound()
   {
      System.out.println(name + " makes a sound");
   }
}
class Dog extends Animal
{
   String breed;
   Dog(String name, String breed)
   {
      super(name);
      this.breed = breed;
      System.out.println("Dog constructor called");
   }
   void sound()
   {
      super.sound();
      System.out.println(name + " barks");
   }
   void displayBreed() {
      System.out.println(name + " is a " + breed);
   }
}
public class SuperKeyword
{
```

```
    public static void main(String[] args)
    {
        Dog dog = new Dog("Buddy", "Golden Retriever");
        dog.sound();
        dog.displayBreed();
    }
}
```

Output

```
Animal constructor called
Dog constructor called
Buddy makes a sound
Buddy barks
Buddy is a Golden Retriever
```

2. Write a JAVA program that implements Runtime polymorphism

```
class Animal
{
   void sound()
   {
      System.out.println("Animal makes a sound");
   }
}
class Dog extends Animal
{
   void sound()
   {
      System.out.println("Dog barks");
   }
}
class Cat extends Animal
{
  void sound()
   {
      System.out.println("Cat meows");
   }
}
public class RuntimePolymarphism
{
   public static void main(String[] args)
   {
      Animal myDog = new Dog();
      Animal myCat = new Cat();

      myDog.sound();
      myCat.sound();
   }
}
```

Output

Dog barks

Cat meows

3. Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

```
interface CanStudy
{
   void study();
}
interface CanParticipate
{
   void participate();
}
interface CanTakeExams
{
   void takeExam();
}
class Student implements CanStudy, CanParticipate, CanTakeExams
{
   private String name;
   public Student(String name)
   {
      this.name = name;
   }
   public void study()
   {
      System.out.println(name + " is studying.");
   }
   public void participate()
   {
      System.out.println(name + " is participating in extracurricular activities.");
   }
   public void takeExam()
   {
      System.out.println(name + " is taking an exam.");
   }
}
public class MultipleInheritance
```

```
{
   public static void main(String[] args)
   {
      Student student = new Student("Alice");

      student.study();
      student.participate();
      student.takeExam();
   }
}
```

Output

Alice is studying.

Alice is participating in extracurricular activities.

Alice is taking an exam.