Week2

Program 1

Aim: Write a JAVA program to given the example for 'super' keyword.

Description:

Source code:

```
class Person
{
	int id;
	String name;
	Person(int id,String name)
	{
		this.id=id;
		this.name=name;
	}
}
class Emp extends Person
{
	float salary;
	Emp(int id,String name,float salary)
	{
		super(id,name);//reusing parent constructor
		this.salary=salary;
	}
	void display()
	{
		System.out.println(id+" "+name+" "+salary);
	}
}
class SuperKeywordEx
{
	public static void main(String[] args)
	{
		Emp e1=new Emp(1,"spkreddy",45000f);
		e1.display();
	}
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>JAVAC SuperKeywordEx.java

C:\Users\SPK REDDY\Desktop\AIDS>JAVA SuperKeywordEx
1 SPKREDDY 45000.0
```

Aim: Write a JAVA program that illustrates simple inheritance.

Program 2

Aim: Write a JAVA program that illustrates simple inheritance.

Description:

Source code:

```
class Library
{
	int id;

	Library(int id)
	{
	this.id = id;
	}
}

class Student extends Library
{
	String name;

	Student(int id, String name)
	{
		super(id);
	this.name = name;
	}
	void display()
	{
	System.out.println("Id = " + id + " Student Name = " +name);
	}
}
class SimpleInheritance
{

	public static void main(String[] args)
	{
		Student s = new Student(1, "spkreddy");
		s.display();
    }
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>javac SimpleInheritance.java

C:\Users\SPK REDDY\Desktop\AIDS>java SimpleInheritance
Id = 1 Student Name = spkreddy
```

Program 3

Aim: Write a JAVA program that illustrates multi-level inheritance.

Source code:

```
class GrandFather
{
	void ownHouse()
	{
		System.out.println("own 2BHK house");
```

```
	}

}
class Father extends GrandFather
{
	void ownLand()
	{
		System.out.println("own 1000 sq. yards land");
	}
}
class Son extends Father
{
	void ownCar()
	{
		System.out.println("own Audi car");
	}

}
class MultilevelInheritance
{
	public static void main(String args[])
	{
		Son obj=new Son();
		obj.ownHouse();
		obj.ownLand();
		obj.ownCar();
	}
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>javac MultilevelInheritance.java

C:\Users\SPK REDDY\Desktop\AIDS>java MultilevelInheritance
own 2BHK house
own 1000 sq. yards land
own Audi car
```

Program 4:
Aim: Write a JAVA program demonstrating the difference between method overloadin
overriding

Source code:

Program 4:

Aim: Write a JAVA program demonstrating the difference between method overloading and method overriding

Source code:

```
//Program for Method loading

class Sample
{
	static void sum(int a, int b)
	{
		System.out.println((a+b));
	}
	static void sum(int a, int b, int c)
	{
```

```
                System.out.println((a+b+c));
        }
        static void sum(float a,float b)
        {
                System.out.println((a+b));
        }
}
class MethodOverloading
{
        public static void main(String args[])
        {
                Sample s=new Sample();
                s.sum(1, 2);
                s.sum(2, 3, 4);
                s.sum(2.3f,3.4f);
        }
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>javac MethodOverloading.java

C:\Users\SPK REDDY\Desktop\AIDS>java MethodOverloading
3
9
5.7
```

//program for method overriding

```
class A
{
        int a,b;
        A(int a, int b)
        {
                this.a = a;
                this.b = b;
        }
        void sum()
        {
                System.out.println("Sum is: "+(a+b));
        }
}
class B extends A
{
        int c;
        B(int a, int b, int c)
        {
                super(a, b);
                this.c = c;
        }
        void sum()
        {
                System.out.println("Sum is: "+(a+b+c));
        }
```

```
}
class MethodOverriding
{
	public static void main(String args[])
	{
		A a = new A(1, 2);
		a.sum();
		B b = new B(5, 6, 7);
		b.sum();
	}
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>javac MethodOverriding.java

C:\Users\SPK REDDY\Desktop\AIDS>java MethodOverriding
Sum is: 3
Sum is: 18
```

Program 5

Aim: Write a JAVA program demonstrating the difference between method overloading and constructor overloading.

Description:

Source code:

```
//Program for constructor overload

class Student
{
	int id;
	String name;
	int fee;
	Student(int i,String n)
	{
		id = i;
		name = n;

	}
	Student(int i, String n, int a)
	{
		id = i;
		name = n;
		fee = a;
	}
	void display()
	{
		System.out.println(id+" "+name+" "+fee);
	}
}
class ConstructorOverload
```

```
{
	public static void main(String args[])
	{
		Student s1 = new Student(111,"Karan");
		Student s2 = new Student(222,"Aryan",25);
		s1.display();
		s2.display();
	}
}
```

```
C:\Users\SPK REDDY\Desktop\AIDS>javac ConstructorOverload.java

C:\Users\SPK REDDY\Desktop\AIDS>java ConstructorOverload
111 Karan 0
222 Aryan 25
```

```
//method overload

class Sample
{
	 void sum(int a, int b)
	{
		System.out.println((a+b));
	}
	 void sum(int a, int b, int c)
	{
		System.out.println((a+b+c));
	}
	void sum(float a,float b)
	{
		System.out.println((a+b));
	}
}
class MethodOverloading
{
	public static void main(String args[])
	{
		Sample s=new Sample();
		s.sum(1, 2);
		s.sum(2, 3, 4);
		s.sum(2.3f,3.4f);
	}
}
```

```
E:\csebjava\week2>javac MethodOverloading.java

E:\csebjava\week2>java MethodOverloading
3
9
5.7
```