Name: Anish Ashok Sharma **Sap id: 60003220045**

Branch: Information Technology **Div: D/IT1**

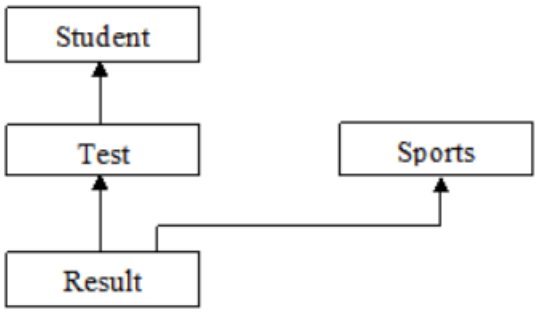
Course: **Object Oriented Programming using Java**

Experiment no. 8

Aim: To implement multiple inheritance using interfaces and method overriding.

Problem Statement 1:

WAP to implement three classes namely Student, Test and Result. Student class has member as rollno, and read(). Test class has members as sem1_marks and sem2_marks and read(). Result class has member as total. Create an interface named sports that has a member score (). Derive Test class from Student and Result class has multiple inheritances from Test and Sports. Total is formula based on sem1_marks, sem2_mark and score.



Code:

```
import java.util.*;

class Student

{

        int roll;

        Scanner sc=new Scanner(System.in);

        public void read()

        {

                System.out.println("Enter roll number:");

                roll=sc.nextInt();

        }

}

class Test extends Student

{

        int sem1,sem2,sportMarks;

        public void read()

        {
```

```
                super.read();
                System.out.println("Enter Sem1 marks:");
                sem1=sc.nextInt();
                System.out.println("Enter Sem2 marks:");
                sem2=sc.nextInt();
        }
}
interface Sports
{
        public abstract void read();
}
class Result extends Test implements Sports
{
        public void read()
        {
                super.read();
                System.out.println("Enter sports marks");
                sportMarks=sc.nextInt();
        }
        public void total()
        {
                System.out.println("Total marks of Anish:"+(sem1+sem2+sportMarks));
        }
}
class EngineerMarks
{
        public static void main(String[] args)
        {
                Result Anish=new Result();
                Anish.read();
                Anish.total();
        }
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>java EngineerMarks
Enter roll number:
11
Enter Sem1 marks:
99
Enter Sem2 marks:
99
Enter sports marks
99
Total marks of Anish:297
```

Problem Statement 2:

Demonstrate that a variable is constant, method cannot be overridden, class cannot be inherited using final keyword

Code:

```
class FinalVar
{
	public static void main(String[] args)
	{
		final int a=89;
		a=2;
		System.out.println(a);
	}
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>javac FinalVar.java
FinalVar.java:6: error: cannot assign a value to final variable a
                a=2;
                ^
1 error
```

Code:

```
final class A
{
	public final void show()
	{
		System.out.println("In A class");
	}
}
class B extends A
{
	public void show()
	{
		System.out.println("In B class");
	}
```

```
}
class FinalClass
{
	public static void main(String[] args)
	{
		B obj=new B();
		obj.show();
	}
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>javac FinalMethod.java
FinalMethod.java:10: error: show() in B cannot override show() in A
        public void show()
                    ^
  overridden method is final
1 error
```

Code:

```
final class A
{
	public final void show()
	{
		System.out.println("In A class");
	}
}
class B extends A
{
	public void show()
	{
		System.out.println("In B class");
	}
}
class FinalClass
{
```

```
        public static void main(String[] args)
        {
                B obj=new B();
                obj.show();
        }
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>javac FinalClass.java
FinalClass.java:8: error: cannot inherit from final A
class B extends A
                ^
FinalClass.java:10: error: show() in B cannot override show() in A
        public void show()
                    ^
  overridden method is final
2 errors
```

Problem Statement 3:

Demonstrate using a suitable example that a base class reference variable can point to a child class object using the concept of dynamic method dispatch.

Code:

```
class A
{
	int a=89;
	public void display1()
	{
		System.out.println("In A class");
	}
}
class B extends A
{
	public void display2()
	{
		System.out.println("In A class");
	}
}
class Dispatch
{
	public static void main(String[] args)
	{
		A obj=new B();
		obj.display1();
		System.out.println(obj.a);
	}
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>javac Dispatch.java

C:\Users\91720\OneDrive\Desktop\Anish Java>java Dispatch
In A class
89
```

Problem Statement 4:

WAP to create an object of a class, delete the same object by calling System. gc () and display a message that the "object has been deleted".

Code:

```
class Circle
{
	public void display()
	{
		System.out.println("This s a circle");
	}
}
class Deletion
{
	public static void main(String args[])
	{
		Circle c=new Circle();
		c.display();
		c=null;
		System.gc();
		System.out.println("Object is deleted");
		//c.display();
	}
}
```

Output

```
C:\Users\91720\OneDrive\Desktop\Anish Java>javac Deletion.java

C:\Users\91720\OneDrive\Desktop\Anish Java>java Deletion
This s a circle
Object is deleted
```