Experiment No.8
Aim: Basic programming constructs like branching and looping
Name: Aman Mehtar
Roll no: 32
Date of Performance: 5/9/24
Date of Submission: 12/9/24

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:

```
import java.io.*;
import java.util.*;
class SingleInheritance{
	public static void main(String args[]){
		Cube sc=new Cube();
		int resultsquare=sc.calculate(17);
		int resultcube=sc.calculate1(17);
		System.out.println("Square is :"+resultsquare);
		System.out.println("Cube is :"+resultcube);
}

}
class operation{
	int calculate(int num){
		int sum=num*num;
		return sum;

}
}
```

```
class Cube extends operation{
        int calculate1(int num){
                int cube=num*num*num;
                return cube;
}
}
```

Output:

```
Square is :289
Cube is :4913

=== Code Execution Successful ===
```

Conclusion:

Single inheritance is a fundamental aspect of object-oriented programming in Java that provides simplicity, clarity, and code reusability. By allowing a subclass to inherit from only one superclass, it avoids the complexities of multiple inheritance while promoting a clear and manageable class hierarchy. While it has some limitations, the benefits it offers make it a preferred choice in many programming scenarios, especially in well-structured software designs.