Name	**Shubhan Singh**
UID no.	**2022300118**
Experiment No.	**3**

PROBLEM STATEMENT :	*Create a Test class with a data double base, int power, int logBase, int argument.* *Create a default, no-argument constructor which sets the default value of all variables to 2.* *There are 2 overloaded functions:* *1. double calculate (double base, int power)* *This function returns the value when *base* is raised to *power** *For example: calculate (3.0, 2) returns the value of 3.0 raised to 2 i.e., 9.0* *2. double calculate (int logBase, int argument)* *This function returns the value of the log of *argument* to the base *logBase**. *For example: calculate (3, 9) returns log of 9 to the base 3 i.e., 2.0* *Create a main method in a separate class to call the above functions with the following inputs:* *1. calculate (2, 4)* *2. calculate (2.0, 4.0)* *Create a display() method which displays the output based on the type of Test object created.*
THEORY:	**Math class:** The Math class in Java is a built-in class that provides a set of mathematical functions and constants that can be used in Java programs. This class includes methods for performing various mathematical operations, such as finding the absolute value, square root, maximum, and minimum of numbers. Two commonly used methods in the Math class are Math.pow() and Math.log(). Math.pow() is a method that takes two arguments, a base and a power, and returns the result of raising the base to the power. The base and power can be any double value. The method returns a double value. For example, calling Math.pow(2, 3) would return 8.0, because 2 raised to the power of 3 is 8. Math.log() is a method that takes one or two arguments, an argument and an

	optional base, and returns the natural logarithm of the argument, or the logarithm of the argument to the specified base. If the base is not specified, the method returns the natural logarithm. The argument must be a positive double value and the base, if specified, must be a positive double value greater than 1. The method returns a double value. For example, calling Math.log(10) would return 2.302585092994046, because the natural logarithm of 10 is approximately 2.302585. **Method overloading in Java:** Method overloading is a feature in Java that allows you to define multiple methods with the same name in a class, but with different parameters. The Java compiler distinguishes between these methods based on the number, order, and types of the parameters in the method signature. When a method is called, the compiler determines which method to execute based on the arguments that are passed.
PROGRAM:	``` import java.lang.Math; // Define a class named Test class Test { // Declare instance variables double base, ans; int power; int log_base, argument; // Define a constructor that sets all instance variables to 2 Test() { base = power = log_base = argument = 2; } // Define a method to calculate base raised to the power of power void Calculate(double base, int power) { ans = Math.pow(base, power); } // Define a method to calculate the logarithm of argument to the base of log_base void Calculate(int log_base, int argument) { ans = Math.log(argument) / Math.log(log_base); } // Define a method to display the value of ans void display() { System.out.println(ans); } } // Define a public class named Sc_calc ```

```java
public class Sc_calc {
    public static void main(String[] args) {
        // Create an instance of the Test class
        Test obj = new Test();

        // Call the Calculate method with a double and an
integer argument
        obj.Calculate(2, 4);
        System.out.print("The value of log 4 base 2 is: ");
        // Display the result
        obj.display();

        // Call the Calculate method with a double and an
integer argument
        obj.Calculate(2.0, 4);
        System.out.print("The value of 2^4 is:");
        // Display the result
        obj.display();
    }
}
```

RESULT:

```
The value of log 4 base 2 is: 2.0
The value of 2^4 is: 16.0

Process finished with exit code 0
```