

Report : Scientific Calculator.

Classes :

1. ArithmeticCalculation
2. TrigonometricCalculation
3. SIP
4. NumberConversion
5. memoryStorage
6. Calculation (Abstract class)
7. calculationHistory
8. Main

ArithmeticCalculation:

This class is responsible for all the Arithmetic calculation and the list of calculation are as follows: 1. Addition 2.Subtraction 3. Division 4. Multiplication 5. Exponent 6. Square root 7.logarithm 8. Natural logarithm

The ArithmeticCalculation class implements an interface named as Arithmetic class which contains all the unimplemented methods inside it which illustrates the concept of **Abstraction** of Java.

Example:

```
interface ArithmeticClass{  
    double addition(double a, double b);  
    double subtraction(double a, double b);  
    double division(double a, double b);  
    double multiply(double a, double b);  
    double exponent(double a, double b);  
    double squareRoot(double a);  
    double log(double a);  
    double nlog(double a);  
}
```

Separate methods are created for each calculation, The ArithmeticCalculation class extends the Calculation class to get and set the private variable, Which demonstrates the concept of **Encapsulation & inheritance** of Java.

The method structure of the calculation inside the ArithmeticCalculation class looks something like this:

```
public double multiply(double a, double b) {  
    double c = a * b;  
    return c;  
  
}
```

In the above code the method is responsible to take in the variable a and b and returning the final result.

This structure is followed by all the method present inside the ArithmeticCalculation.

Inside the ArithmeticCalculation there are two method with same name but one is with parameter and the other one without parameter.

When the main method gets executed the method with the parameter is called which demonstrates understanding of **Method Overloading** which is ultimately falls under **Polymorphism**

TrigonometriCalculation class follows the same structure as the ArithmeticCalculation class, That means all the method are assigned with only one calculation, which represents the understanding of SOLID i.e the understanding of SRP (**Single Responsibility Principle**).

The Interface implemented by the TrigonometricCalcualtion classs is represented below,

```
interface trigoCalculation {  
    double sine(double a);  
  
    double cosine(double a);  
  
    double tangent(double a);  
  
    double sinInverse(double a);  
  
    double cosInverse(double a);  
  
    double tanInverse(double a);  
  
}
```

The rest of the class follows the same design pattern as the classes explained above.