

## Calculator\_f:

**Run: Calculator\_f.main(null);**

```
public class Calculator_f {

    public static Integer add(Integer a, Integer b) {
        return a + b;
    }

    public static Integer sub(Integer a, Integer b) {
        return a - b;
    }

    public static Integer mul(Integer a, Integer b) {
        return a * b;
    }

    public static Double div(Integer a, Integer b) {
        return (Double)a / b;
    }

    public static Double power(Integer base, Integer exponent) {
        return Math.pow(base, exponent);
    }

    public static Double squareRoot(Integer a) {
        return Math.sqrt(a);
    }

    public static Integer modulo(Integer a, Integer b) {
        return Math.mod(a,b);
    }

    public static void main(String[] args) {
        Integer a = 10;
        Integer b = 20;

        System.debug('Addition is: ' + add(a, b));
        System.debug('Subtraction is: ' + sub(a, b));
        System.debug('Multiplication is: ' + mul(a, b));
        System.debug('Division is: ' + div(a, b));
        System.debug('Power is: ' + power(a, b));
    }
}
```

```
        System.debug('Square Root of ' + a + ' is: ' + squareRoot(a));
        System.debug('Modulo is: ' + modulo(a, b));
    }
}
```

## GreatestNumber

**Run: GreatestNumber.main(null);**

```
public class GreatestNumber {

    public static Integer getMax(Integer a, Integer b, Integer c){
        return Math.max(a,Math.max(b,c));
    }

    public static Integer findMax(Integer num1, Integer num2, Integer num3) {

        Integer max = num1;

        if (num2 > max) {
            max = num2;
        }

        if (num3 > max) {
            max = num3;
        }

        return max;
    }

    public static void main(String[] args){
        System.debug('Maximum is: ' + getMax(1,3,2));
        System.debug('Maximum is: ' + findMax(1,3,2));
    }
}
```

## Electricity Bill and Temperature

**Run: ElectricityBill\_Temperature .main(null);**

```
public class ElectricityBill_Temperature {

    public static Decimal calculateElectricityBill(Integer unitsConsumed, Decimal
ratePerUnit) {
        return unitsConsumed * ratePerUnit;
    }

    public static Decimal convertCelsiusToFahrenheit(Decimal celsiusTemperature) {
        return (celsiusTemperature * 9/5) + 32;
    }

    public static void main(String[] args) {

        System.debug('Bill is: ' + calculateElectricityBill(400,10));
        System.debug('Fahrenheit is: ' + convertCelsiusToFahrenheit(20));
    }
}
```

## CurrencyConverter

**Run: CurrencyConverter.main(null);**

```
public class CurrencyConverter {

    public static Decimal convertCurrency(Decimal amount, String fromCurrency, String
toCurrency) {

        Map<String, Decimal> exchangeRates = new Map<String, Decimal>{
            'USD-EUR' => 0.85, // 1 USD = 0.85 EUR
            'USD-GBP' => 0.72, // 1 USD = 0.72 GBP
            'USD-INR' => 75.00, // 1 USD = 75.00 INR
            'EUR-USD' => 1.18, // 1 EUR = 1.18 USD
            'EUR-GBP' => 0.85, // 1 EUR = 0.85 GBP
            'EUR-INR' => 88.00, // 1 EUR = 88.00 INR
        };
    }
}
```

```
'GBP-USD' => 1.39, // 1 GBP = 1.39 USD
'GBP-EUR' => 1.18, // 1 GBP = 1.18 EUR
'GBP-INR' => 94.00, // 1 GBP = 94.00 INR
'INR-USD' => 0.013, // 1 INR = 0.013 USD
'INR-EUR' => 0.011, // 1 INR = 0.011 EUR
'INR-GBP' => 0.011 // 1 INR = 0.011 GBP
};
```

```
String exchangeKey = fromCurrency + '-' + toCurrency;
if (exchangeRates.containsKey(exchangeKey)) {
```

```
    Decimal exchangeRate = exchangeRates.get(exchangeKey);
    Decimal convertedAmount = amount * exchangeRate;
    return convertedAmount;
```

```
    } else {
        System.debug('Conversion rates not available for the selected currencies.');
```

```
        return null;
    }
}
```

```
public static void main(String[] args){
```

```
    Decimal amount = 100;
    String fromCurrency = 'USD';
    String toCurrency = 'EUR';
```

```
    Decimal convertedAmount = convertCurrency(amount, fromCurrency, toCurrency);
    if (convertedAmount != null) {
        System.debug('Converted amount: ' + convertedAmount + ' ' + toCurrency);
```

```
    }
}
}
```

## StudentMarkSheet

**Run: StudentMarksheet .main(null);**

```
public class StudentMarksheet {

    public class Marksheet {
        public String studentName { get; set; }
        public Integer rollNumber { get; set; }
        public Map<String, Integer> subjectMarks { get; set; }
        public Integer totalMarks { get; set; }
        public Decimal percentage { get; set; }

        public Marksheet(String name, Integer roll) {
            studentName = name;
            rollNumber = roll;
            subjectMarks = new Map<String, Integer>();
        }

        public void addMarks(String subject, Integer marks) {
            subjectMarks.put(subject, marks);
        }

        public void calculateTotalAndPercentage() {
            totalMarks = 0;
            for(Integer marks : subjectMarks.values()) {
                totalMarks += marks;
            }
            percentage = (Decimal)totalMarks / (subjectMarks.size() * 100) * 100;
        }

        public void printMarksheet() {
            System.debug('Student Name: ' + studentName);
            System.debug('Roll Number: ' + rollNumber);
        }
    }
}
```

```

        System.debug('-----');
        for(String subject : subjectMarks.keySet()) {
            System.debug(subject + ': ' + subjectMarks.get(subject));
        }
        System.debug('-----');
        System.debug('Total Marks: ' + totalMarks);
        System.debug('Percentage: ' + percentage + '%');
    }
}

```

```

public static void main() {

```

```

    Marksheet marksheet = new Marksheet('John Doe', 101);

```

```

    // Add marks for different subjects
    marksheet.addMarks('Maths', 90);
    marksheet.addMarks('Science', 85);
    marksheet.addMarks('English', 80);
    marksheet.addMarks('History', 75);

```

```

    // Calculate total marks and percentage
    marksheet.calculateTotalAndPercentage();

```

```

    // Print marksheet
    marksheet.printMarksheet();

```

```

    }
}

```

## Array Example

```
public class Practice {
    public static integer electric_bill(integer unit, integer ratePerUnit){
        return unit * ratePerUnit;
    }
    public static double temperatureConverter(integer tempC){
        return (double)(tempC*9/5)+32;
    }
    public static integer fib(integer n) {
        if (n == 0) {
            return 0;
        }
        if (n == 1) {
            return 1;
        }
        return fib(n - 1) + fib(n - 2);
    }
    public static void main(String []args){
        System.debug('Electricity bill : '+electric_bill(40,10));
        System.debug('40 C int Fahrehnite : '+temperatureConverter(40));
        string []numbers = new string[10];

        for(integer i = 0; i < 10; i++){
            numbers[i] = fib(i) + '->';
        }
        for(integer i = 0; i < 10; i++){
            System.debug(numbers[i]);
        }
    }
}
```