Salesforce Apex Programming Practical

Mathematical Calculator:

```
public class Calculator {
   public static void add(Decimal num1, Decimal num2) {
      Decimal ans=num1+num2;
      System.debug('Addition of num1+num2:'+ans);
   }
   public static void subtract(Decimal num1, Decimal num2) {
      Decimal ans=num1-num2;
      System.debug('Subtraction of num1-num2:'+ans);
   }
   public static void multiply(Decimal num1, Decimal num2) {
      Decimal ans=num1*num2;
      System.debug('Multiplication of num1*num2:'+ans);
   }
   public static void divide(Decimal num1, Decimal num2) {
      if (num2 == 0) {
         System.debug('Division by zero is not allowed.');
      }
      Decimal ans=num1/num2;
      System.debug('Division of num1/num2:'+ans);
   }
         public static void exponentiation(Integer base, Integer exponent) {
     Decimal ans=Math.pow(base,exponent);
      System.debug('Exponentitation of num1**num2:'+ans);
   }
         public static void modulo(Integer num1, Integer moduloBy) {
         if (moduloBy == 0) {
         System.debug('Division by zero is not allowed.');
         }
         Integer ans= Math.mod(num1, moduloBy);
```

```
        System.debug('Modulo of num1 % moduloBy:'+ans);
       }
    public static void square(Integer num) {
        Decimal ans =num * num;
        System.debug('Square of number is :'+ans);
    }
    public static void squareRoot(Decimal num) {
        if (num < 0) {
            System.debug('Square root of negative number is not allowed.');
        }
        Decimal ans= Math.sqrt(num);
        System.debug('Square Root of num is:' +ans);
    }
}
```

Student Marksheet Generator:

```
public class StudetMarkSheet {
  public static void generateMarkSheet(String studentName, Integer englishM, Integer mathM,
Integer scienceM) {
    Decimal totalM = mathM + englishM + scienceM;
    Decimal percent = (totalM / 300) * 100;
    String grade;
    if (percent >= 90) {
        grade = 'A+';
    } else if (percent >= 80) {
        grade = 'A';
    } else if (percent >= 70) {
        grade = 'B';
    } else if (percent >= 60) {
        grade = 'C';
    } else if (percent >= 50) {
```

```
        grade = 'D';
    } else {
        grade = 'F';
    }
    System.debug('Student Name: ' + studentName);
    System.debug('English Marks: ' + englishM);
    System.debug('Math Marks: ' + mathM);
    System.debug('Science Marks: ' + scienceM);
    System.debug('Total Marks Out of 300: ' + totalM);
    System.debug('Percentage: ' + percent.setScale(2));
    System.debug('Grade: ' + grade);
}

}
```

Greatest Three Number:

```
public class ThreeNumber {
    public static void findGreatest(Decimal num1,Decimal num2,Decimal num3){
        if (num1 == num2 && num2 == num3) {
            System.debug('All numbers are equal.');
        }
        else{
        Decimal ans=num1;
        if(num2>ans){
            ans=num2;
        }
        if(num3>ans){
            ans=num3;
        }
        System.debug('Greatest among All three Numbers is : '+ans);
        }
```

```
    }
}
```

Celsius to Fahrenheit converter:

```
public class Temperature {
    //Celsius to Fahrenheit =(Celcius * 9/5) + 32
    //Fahrenheit to Celcius =(Fahrenheit - 32) * 5/9;
    public static void CelsiusToFahrenheit(Decimal Celsius) {
        Decimal ans = (Celsius * 9/5) + 32;
        System.debug('Celsius: ' + Celsius + '°C = Fahrenheit: ' + ans + '°F');
    }

    public static void FahrenheitToCelsius(Decimal Fahrenheit) {
        Decimal ans = (Fahrenheit - 32) * 5/9;
        System.debug('Fahrenheit: ' + Fahrenheit + '°F = Celsius: ' + ans + '°C');
    }
}
```

Currency Converter:

```
public class CurrencyConverter {
    // Exchange rates (as of current rates)
    private static Decimal USD_TO_EUR_RATE = 0.85; // 1 USD = 0.85 EUR
    private static Decimal EUR_TO_USD_RATE = 1.18; // 1 EUR = 1.18 USD
    private static Decimal USD_TO_INR_RATE = 74.5; // 1 USD = 74.5 INR
    private static Decimal INR_TO_USD_RATE = 0.013; // 1 INR = 0.013 USD

    // Convert from USD to EUR
    public static void convertUSDToEUR(Decimal usdAmount) {
                    Decimal eurAmount =usdAmount * USD_TO_EUR_RATE;
        System.debug(eurAmount);
    }
```

```
    // Convert from EUR to USD
    public static Decimal convertEURToUSD(Decimal eurAmount) {
        return eurAmount / EUR_TO_USD_RATE;
    }

    // Convert from USD to INR
    public static Decimal convertUSDToINR(Decimal usdAmount) {
        return usdAmount * USD_TO_INR_RATE;
    }

    // Convert from INR to USD
    public static void convertINRToUSD(Decimal inrAmount) {
        Decimal USDAmount =inrAmount * INR_TO_USD_RATE;
        System.debug(USDAmount);
    }
}
```

Electricity Bill:

```
public class BillGenerator {
    public static void generate(Integer unit){
        Decimal BillAmount=0;
        Decimal Rate1=5;
        Decimal Rate2=7;
        Decimal Rate3=10;
        if(unit<=100){
            BillAmount=Rate1*unit;
        }
        else if(unit<=200){
            BillAmount=(100*Rate1)+((unit-100)*Rate2);
        }
```

```
        else{
            BillAmount=(100*Rate1)+(100*Rate2)+((unit-200)*Rate3);
        }
        System.debug('Unit Consumed:' +unit);
        System.debug('Bill Amount: '+BillAmount);
    }
}
```