Queries – Exercise

Practice

1. Write a definition of a method **norm** that has three double parameters, *x*, *y*, and *z*. The method should return, as a double value, the value of the expression $[x^2 + y^2 + z^2]^{0.5}$.

```
                double power = norm(5,5,5);
                System.out.println(power);

        }
        public static double norm (double x, double y, double z) {
                double expression = Math.pow(Math.pow(x,2) + Math.pow(y,2)
+ Math.pow(z,2),0.5);
                return expression;
        }
```

2. Write a Boolean-valued method **isSquare** with a single int parameter, *n*. The method should return the value true if and only if *n* is the square of some integer.

```
                boolean square;
                square = (isSquare (25));
                System.out.println(square);

        }
        public static boolean isSquare (int n) {
                String str = String.valueOf(n);
                int length = str.length()-1;
                if (str.charAt(length) == '0' || str.charAt(length) == '1'
|| str.charAt(length) == '4' || str.charAt(length) == '5' ||
str.charAt(length) == '6' || str.charAt(length) == '9') {
                        return true;

                }
                else {
                        return false;
                }

        }
```

3. Write a method called **onesDigit** with a single int parameter, *n*. This method will return the last digit (the ones digit) of *n* as a single integer.

```
                int digit = (onesDigit(12345));
                System.out.println(digit);

        }
        public static int onesDigit(int n) {
                String str = String.valueOf(n);
                int length = str.length();
                char lastDigitchar = str.charAt(length-1);
                String  lastDigitString = String.valueOf(lastDigitchar);
                int lastDigit = Integer.parseInt(lastDigitString);
                return lastDigit;
```

```
}
```

4. public static int digit (int n, int position) – will determine and return the value of the digit that is *n* position places from the right stating at 0. For example:
   a. digit (123, 0) should return 1
   b. digit (28457, 2) should return 4
   c. digit (48, 6) should return -1

```
        int position1 = digit (123,0);
        System.out.println(position1);

}
public static int digit (int digit, int n) {
        String str = String.valueOf(digit);
        String str1 = String.valueOf(str.charAt(n));
        int position = Integer.parseInt(str1);
        return position;
}
```

5. Write a character-valued method **convertGrade** that receives a parameter *mark* of type int. This method will return the corresponding letter grade to that mark out of 100. If the mark is out of range return ‘*x*’.

```
        String mark = convertGrade(50);
        System.out.println(mark);


}
public static String convertGrade (int mark) {
        if (mark>=80 && mark<=100 && mark>=0 && mark<=100) {
                String grade = "A";
                return grade;
        }
        else if (mark>=65 && mark<=79 && mark>=0 && mark<=100) {
                String grade = "B";
                return grade;
        }
        else if (mark>=55 && mark<=64 && mark>=0 && mark<=100) {
                String grade = "C";
                return grade;
        }
        else if (mark>=50 && mark<=54 && mark>=0 && mark<=100) {
                String grade = "D";
                return grade;
        }
        else if (mark>=0 && mark<=49 && mark>=0 && mark<=100) {
                String grade = "F";
                return grade;
        }
        else {
                return "x";
        }
        }
```

Main Program - Create a program that simulates a cash register. This cash register will continually offer the following options until 4 is selected:

1 – Add an item

The program will prompt for the cost of an item and add it to the total

2 – Display Tax

The program will display the tax of the current total.

3 – Set discount

The program will prompt the user for a discount as a percentage that is >0% and <100%. The program will calculate and display the amount of discount

4 – Calculate final total

The program will calculate the final cost after discounts and taxes. The program will exit after the final costs have been displayed

Methods - Create the following methods for this program.

a. public static double AddItem(double item, double total) – receives parameters *item* of type double and *total* of type double. AddItem will calculate and return the sum.

b. public static double CalculateTax (double t) – receives parameter *t* of type double that represents the total amount purchase so far. CalculateTax will calculate the tax (HST=13%) of *t*. This method will calculate and return the amount of tax.

c. Public static double CalculateDiscount(double t, double discount) – receives a parameters *t* of type double which represents total amount of the purchases so far and *discount* of type double which represents the discount as a decimal (between 0 and 1). This method will check to see if *discount* is between 0 and 1 and then determine and return the amount of the discount on the purchases. If *discount* is not between 0 and 1, no discount is calculated and -1 is returned.

```
Scanner input = new Scanner (System.in);
        int option;
        do {
            Option();
            option = input.nextInt();
            if (option==1) {
                System.out.println("Enter the cost of the item:");
                double item = input.nextDouble();
                System.out.println("Enter the total amount :");
                double total = input.nextDouble();
                System.out.println("The total is $" +
AddItem(item,total));
            }
            else if (option==2){
                System.out.println("Enter the total amount purchase
so far:");
                double total = input.nextDouble();
                System.out.println ("Tax: $" + CalculateTax(total));
            }
            else if (option==3) {
                System.out.println("Enter the total amount purchase
so far:");
                double total = input.nextInt();
                System.out.println("Enter discount you want:");
                double discount = input.nextDouble();
                System.out.println("Discount: $" +
CalculateDiscount(total,discount));
            }
        }
        while (option!=4);{
            System.out.println("Enter the total amount purchase:");
            double total = input.nextInt();
            System.out.println("Enter discount you want:");
            double discount = input.nextDouble();
            System.out.println("Final total: $" +
CalculateFinalTotal(total, discount));
        }

    }
    public static void Option () {
        System.out.println("1 - Add an item");
        System.out.println("2 - Display Tax ");
        System.out.println("3 - Set discount");
        System.out.println("4 - Calculate final total");
        System.out.println("Enter option:");
    }

    public static double AddItem (double item, double total) {
        double sum = item + total;
        return sum;
    }

    public static double CalculateTax (double t) {
        double HST = 1.13;
        double tax = t*HST;
        return tax;
    }

    public static double CalculateDiscount(double t, double discount) {
        String discountString = Double.toString(discount);
        double length = discountString.length();
        if (length>=0 || length<=1) {
            double discountPurchase = t*discount;
            return discountPurchase;
        }
        else {
            double noDiscount = -1;
            return noDiscount;
        }
    }

    public static double CalculateFinalTotal(double total, double discount)
{
        double finalTotal = ((total-(total*discount))*1.13);
        return finalTotal;
    }
```