Hands-on Experiment for Lecture 7 Method: Worksheet

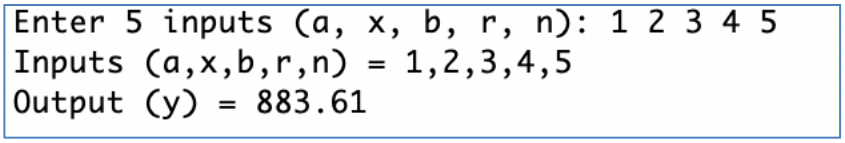
Section________1_______ Date_______________22/9/2020______________________

Student ID ____________6338110221________________________ Name___Nonthapat Kaewamporn______________

Part A: Basic Method

- You are given the following equation:
  - $y = (ax - \ln(\mathrm{b}))r + (\sum_{i=1}^{n}(-1)^{n+1}i^2))\ r^2$

Write a program (TwoTermsMethod.java) that takes (from keyboard) the value of a, b, n, x, r (**all variables are of type int and all calculation must result in double**) then computes **y**. Finally, you must call a method to print all inputs and output (y).



```
Enter 5 inputs (a, x, b, r, n): 1 2 3 4 5
Inputs (a,x,b,r,n) = 1,2,3,4,5
Output (y) = 883.61
```

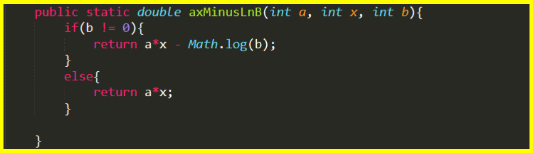
Hint: We can write this program by:

- Write a method that calculates $(ax - \ln(\mathrm{b}))$. Also, when b is 0, the result is just ax
- Write a method that calculates $(\sum_{i=1}^{n}(-1)^{n+1}i^2))$, using the method we defined earlier.
- Write a method that calculates y that calls both previous methods
- Write a method to print all inputs (a, b, n, x, r) and output (y), where *y* must be displayed in two decimal points.

The method that calculates $(ax - \ln(\mathrm{b}))$ has the following header:

```
public static double axMinusLnB (int a, int x, int b)
```

List the source code of the method here:



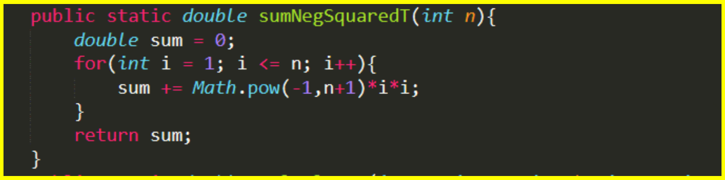
```
public static double axMinusLnB(int a, int x, int b){
    if(b != 0){
        return a*x - Math.log(b);
    }
    else{
        return a*x;
    }

}
```

The method that calculates $(\sum_{i=1}^{n}(-1)^{n+1}i^{2}))$ has the following header:

```
public static double sumNegSquaredI(int n)
```

List the source code of the method here:

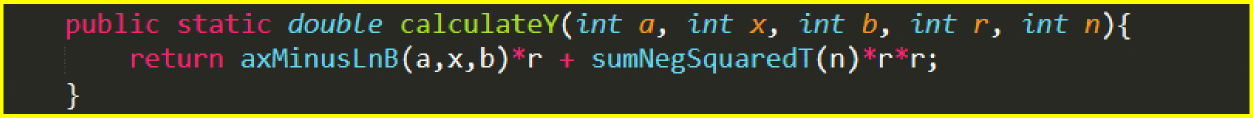


```
public static double sumNegSquaredT(int n){
    double sum = 0;
    for(int i = 1; i <= n; i++){
        sum += Math.pow(-1,n+1)*i*i;
    }
    return sum;
}
```

The method that calculates the value of y has the following header:

```
public static double calculateY(int a, int x, int b, int r, int n)
```

List the source code of the method here:

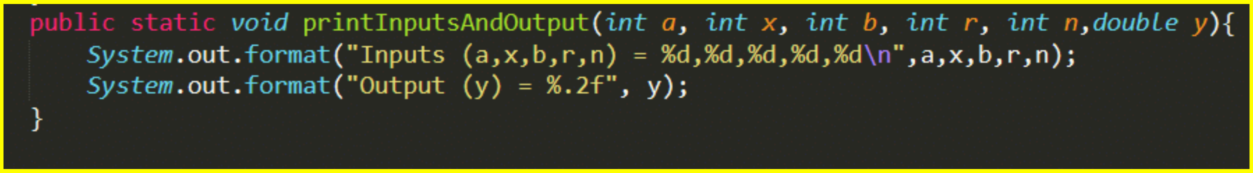


```
public static double calculateY(int a, int x, int b, int r, int n){
    return axMinusLnB(a,x,b)*r + sumNegSquaredT(n)*r*r;
}
```

The method that prints all inputs and output the following header:

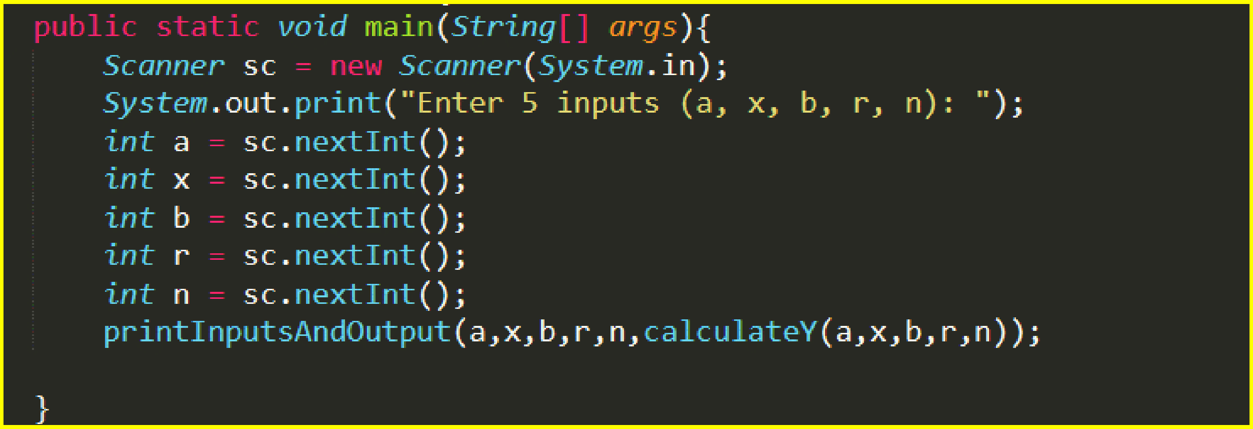
```
public static void printInputsAndOutput(int a, int x, int b, int r, int n, double y)
```

List the source code of the method here:



```
public static void printInputsAndOutput(int a, int x, int b, int r, int n,double y){
    System.out.format("Inputs (a,x,b,r,n) = %d,%d,%d,%d,%d\n",a,x,b,r,n);
    System.out.format("Output (y) = %.2f", y);
}
```

List the source code of your **main** method below:



```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter 5 inputs (a, x, b, r, n): ");
    int a = sc.nextInt();
    int x = sc.nextInt();
    int b = sc.nextInt();
    int r = sc.nextInt();
    int n = sc.nextInt();
    printInputsAndOutput(a,x,b,r,n,calculateY(a,x,b,r,n));

}
```

Write the result of your program in the table below:

a	x	b	r	n	y
1	2	3	4	5	883.61
1	4	2	5	6	-2258.47
2	5	1	3	7	1290.00
3	6	0	7	5	2821.00

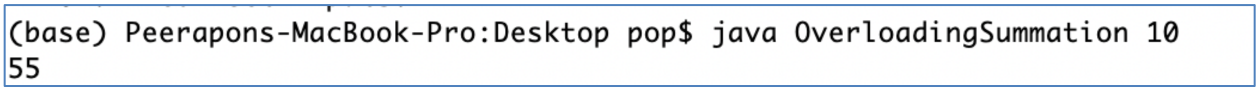
Part B: Overloading Method

Create a java program (OverloadingSum.java) to create three overloading sum methods as follows:

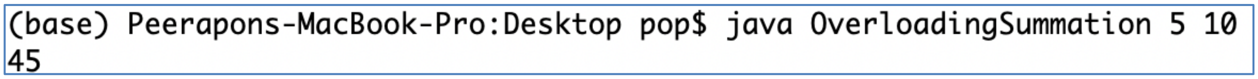
1. int sum(int end); → return a summation from 1 to *end* with an increment by 1
2. int sum(int start, int end); → return a summation from *start* to *end* with an increment by 1
3. int sum(int start, int end, int step); → return a summation from *start* to *end* with an increment by *step*
4. int sum(int start, int end, int step, int bias); → It calculates similarly to the previous option, and then topup (add) with *bias*.

In the main method, the program must check the number of input argument to call the proper summation method. Also, if the number of inputs is incorrect or inputs are NOT integer (NumberFormatException), the program should print “"ERROR: incorrect inputs!"”.

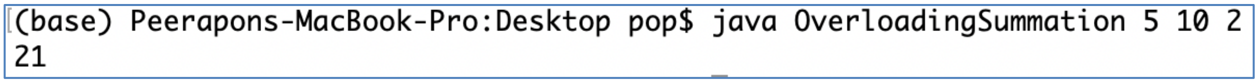
Hint: args.length returns the number of arguments (int) and Integer.parseInt(String) should be helpful. The followings are output examples.



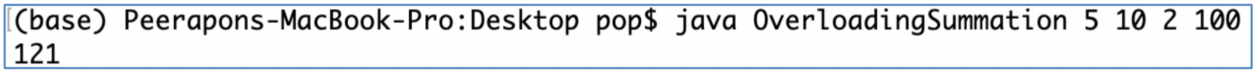
```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation 10
55
```



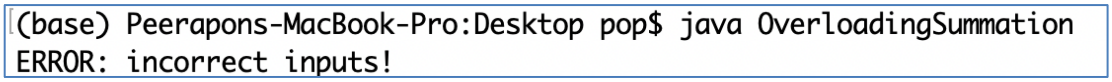
```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation 5 10
45
```



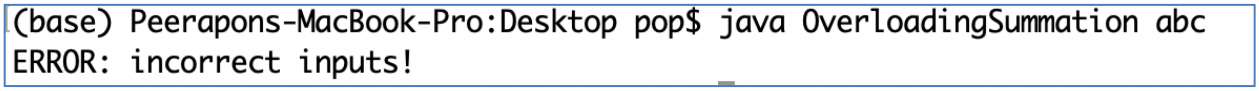
```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation 5 10 2
21
```



```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation 5 10 2 100
121
```

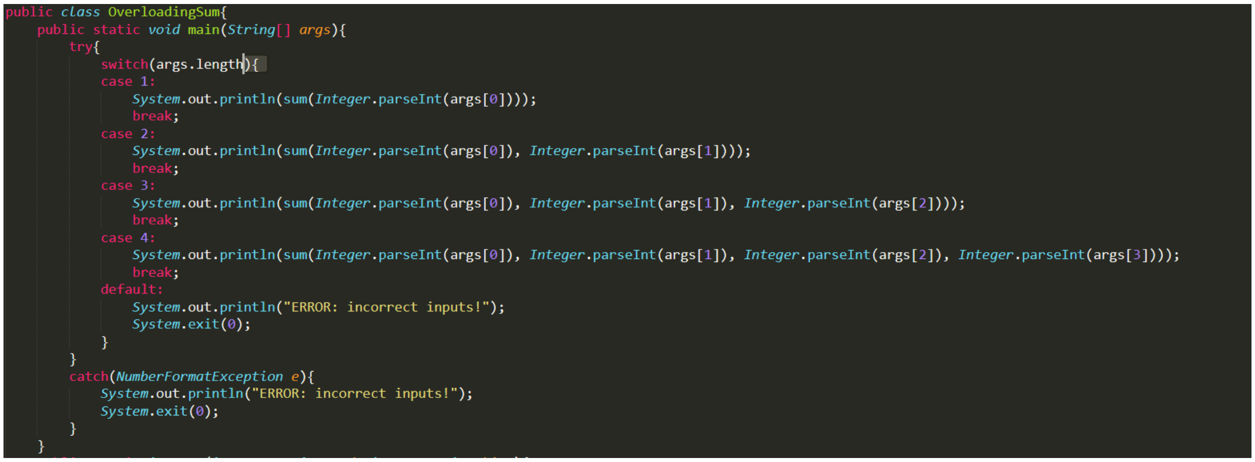


```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation
ERROR: incorrect inputs!
```

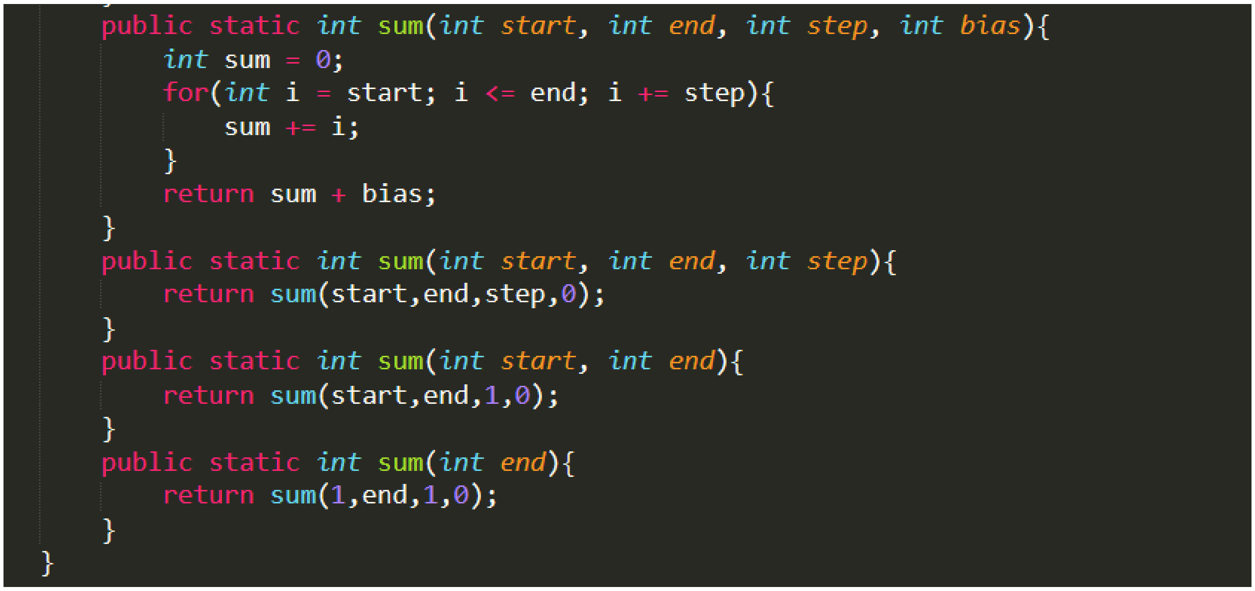


```
(base) Peerapons-MacBook-Pro:Desktop pop$ java OverloadingSummation abc
ERROR: incorrect inputs!
```

List the source code of the method here:



```java
public class OverloadingSum{
    public static void main(String[] args){
        try{
            switch(args.length){
            case 1:
                System.out.println(sum(Integer.parseInt(args[0])));
                break;
            case 2:
                System.out.println(sum(Integer.parseInt(args[0]), Integer.parseInt(args[1])));
                break;
            case 3:
                System.out.println(sum(Integer.parseInt(args[0]), Integer.parseInt(args[1]), Integer.parseInt(args[2])));
                break;
            case 4:
                System.out.println(sum(Integer.parseInt(args[0]), Integer.parseInt(args[1]), Integer.parseInt(args[2]), Integer.parseInt(args[3])));
                break;
            default:
                System.out.println("ERROR: incorrect inputs!");
                System.exit(0);
            }
        }
        catch(NumberFormatException e){
            System.out.println("ERROR: incorrect inputs!");
            System.exit(0);
        }
    }
```



```
    public static int sum(int start, int end, int step, int bias){
        int sum = 0;
        for(int i = start; i <= end; i += step){
            sum += i;
        }
        return sum + bias;
    }
    public static int sum(int start, int end, int step){
        return sum(start,end,step,0);
    }
    public static int sum(int start, int end){
        return sum(start,end,1,0);
    }
    public static int sum(int end){
        return sum(1,end,1,0);
    }
}
```