Daily Work Practice: Date: 9/8/2023

Problem: Basic Calculator

```
package com.ishwarchavan;

import java.util.Scanner;

public class BasicCalculator {

	public static void main(String [] args) {   //main program started
		double num1;          //declaring variables
		double num2;
		double ans;
		char op;

		Scanner reader = new Scanner(System.in);    //taking input from user

		System.out.println("Enter two number: ");

		num1 = reader.nextDouble();
		num2= reader.nextDouble();

		System.out.println("Enter an operator(+, -, *, /,%): ");    //enter
operator sign
		op = reader.next().charAt(0); //taking operator sign ad store in op
variables

		switch(op) {//checking the operator sign with cases and if satisfied
then break executing it
		case '+': ans = num1 + num2;
		break;
		case '-': ans = num1 - num2;
		break;
		case '*': ans = num1 * num2;
		break;
		case '/': ans = num1 / num2;
		break;
		case '%': ans = num1 % num2;
		break;

		default : System.out.println("Error! Enter correct operator");
	    //if user enter unmatch sign then this will print
		return;
		}
		System.out.println("The result is given below: ");
		System.out.println(num1 + " " + op + num2 + "=" + ans);
	}

}
```

Problem: Next Permutation

```
package com.ishwarchavan;

import java.util.Arrays;

public class nextPermutation {
```

```
      public static int[] swap(int nums[], int left,int right) {        //Swap Function
is created with int parameter

            int temp =nums[left];   //swaping data and storing it into  temp variable
            nums[left]= nums[right];
            nums[right]= temp;

      return nums;                          //returning the new array
}
   public static int[] reverse(int nums[], int left, int right) {   //reverse function
is created for reverse value left to right

         while(left<right) {
               int temp = nums[left];                 //reversing array
               nums[left++] = nums[right];
               nums[right- -1]= temp;
         }
         return nums;         //new array return here
         }

    public static boolean firstNextPermutation(int nums[]) {
      //firstnextpermutation function is created for next permutation
      if(nums.length<=1)                   //if this condition is not satisfied then
next permutation is not possible
            return false;
      int last= nums.length-2;

      while(last >=0) {             //if satisfied the condition then break the loop
otherwise decrement
            if(nums[last]< nums[last+1]) {
                  break;
            }
            last--;
      }
      if(last<0)              //if satisfied then return false it's mean no higher
order permutation
            return false;

      int nextGreater = nums.length - 1;

      for (int i =nums.length-1; i>last; i--) {       //loop iterating for nextGreater
            if(nums[i]>nums[last]) {
                  nextGreater =i;
                  break;
            }
      }
      nums= swap(nums, nextGreater, last);
      nums=reverse (nums,last+1, nums.length -1);           //return true as the next
permutation
      return true;
    }
      public static void main(String[] args) {        //main program is started
                  int nums[]= {1,1,5};

                  if(! firstNextPermutation(nums))          //if else condition for
check the higher order or not
                        System.out.println("There is no highrt"+"order permutatio
"+"for the given nums.");
                  else {
                        System.out.println(Arrays.toString(nums));
                  }
               }
            }
```

kh