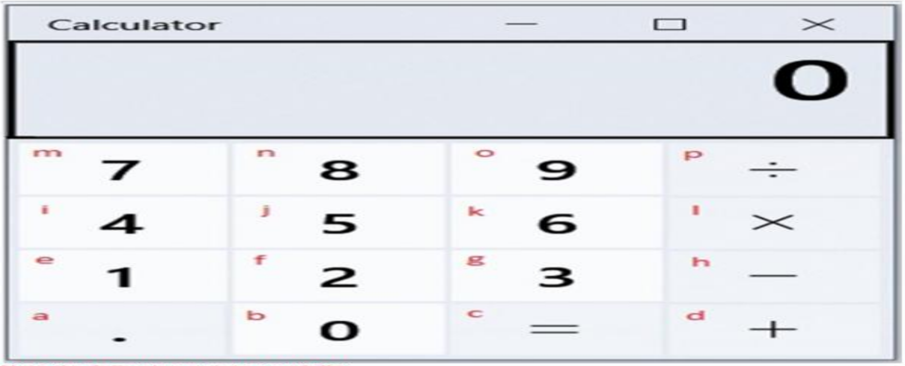
Week 09 : Programming Assignment 4

Due on 2025-03-27, 23:59 IST

Complete the code to **develop an ADVANCED CALCULATOR that emulates all the functions of the GUI Calculator as shown in the image.**



Note the following points carefully:

1. Use only double datatype to store all numeric values.
2. Each button on the calculator should be operated by typing the characters from 'a' to 'p'.
3. To calculate 25-6, User should input fjhkc (where, f for 2, j for 5, h for '-', k for 6 and c for '=').
3. You may use the already defined function gui_map(char).
4. Without '=', operations won't give output as shown in Input_2 and Output_2 example below.
5. The calculator should be able to perform required operations on two operands as shown in the below example:

Input_1:

```
klgc
```

Output_1:

```
18.0
```

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	fghgc	20.0	20.0\n	Passed

The due date for submitting this assignment has passed.
1 out of 1 tests passed.
You scored 100.0/100.

Assignment submitted on 2025-03-25, 21:15 IST

Your last recorded submission was :

```java
import java.util.Scanner;
public class Question94{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine();
sc.close();

        StringBuilder num1 = new StringBuilder();
        StringBuilder num2 = new StringBuilder();
        char operator = ' ';
        boolean hasOperator = false;


        for (char ch : input.toCharArray()) {
            char mappedChar = gui_map(ch);

            if (mappedChar >= '0' && mappedChar <= '9') {
                if (!hasOperator) {
                    num1.append(mappedChar);
                } else {
                    num2.append(mappedChar);
                }
            } else if (mappedChar == '+' || mappedChar == '-' || mappedChar == 'X' || mappedChar == '/') {
                operator = mappedChar;
                hasOperator = true;
            } else if (mappedChar == '=') {
                break;
            }
        }


        double number1 = Double.parseDouble(num1.toString());
        double number2 = Double.parseDouble(num2.toString());
        double result = 0.0;


        switch (operator) {
            case '+': result = number1 + number2; break;
            case '-': result = number1 - number2; break;
            case 'X': result = number1 * number2; break;
            case '/': result = number2 != 0 ? number1 / number2 : Double.NaN; break;
            default: System.out.println("Invalid operation!"); return;
        }


        System.out.println(result);
}// The main() method ends here.

// A method that takes a character as input and returns the corresponding GUI character
    static char gui_map(char in){
        char out = 'N';// N = Null/Empty
        char gm[][]={{'a','.'}
                    ,{'b','0'}
                    ,{'c','='}
                    ,{'d','+'}
                    ,{'e','1'}
                    ,{'f','2'}
                    ,{'g','3'}
                    ,{'h','-'}
                    ,{'i','4'}
                    ,{'j','5'}
                    ,{'k','6'}
                    ,{'l','X'}
                    ,{'m','7'}
                    ,{'n','8'}
                    ,{'o','9'}
                    ,{'p','/'}};

        // Checking for maps
        for(int i=0; i<gm.length; i++){
            if(gm[i][0]==in){
                out=gm[i][1];
                break;
            }
        }
        return out;
    }
}
```

Sample solutions (Provided by instructor)

```java
import java.util.Scanner;
public class Question94{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine();
char seq[] = input.toCharArray();
        int outflag=0;

        // Start the mapping process for each input character
        for(int i=0; i<seq.length; i++){
            seq[i]=gui_map(seq[i]);
        }

        //Print Mapped GUI (remove comment to see the mapped sequence input)
        /*
        for(int i=0; i<seq.length; i++){
            System.out.print(seq[i]);
        }
        */

        // Use double type of values for entire calculation
        double operand1=0.0;
        String o1="";
        double operand2=0.0;
        String o2="";
        double output=0.0;

        // Perform calculaton operations
        outerloop:
        for(int i=0; i<seq.length; i++){
            int r=0;
            if(seq[i]=='+'||seq[i]=='-'||seq[i]=='/'||seq[i]=='X'||seq[i]=='='){
                for(int j=0; j<i; j++){
                    o1+=Character.toString(seq[j]);
                }
                operand1=Double.parseDouble(o1);
                for(int k=i+1; k<seq.length; k++){
                    if(seq[k]=='='){
                        outflag=1;
                        operand2=Double.parseDouble(o2);
                        if(seq[i]=='+'){
                            output=operand1+operand2;
                        }else if(seq[i]=='-'){
                            output=operand1-operand2;
                        }else if(seq[i]=='/'){
                            output=operand1/operand2;
                        }else if(seq[i]=='X'){
                            output=operand1*operand2;
                        }
                        break outerloop;
                    }else{
                        o2+=Character.toString(seq[k]);
                    }
                }
            }
        }

        // Check if output is available and print the output
        if(outflag==1)
            System.out.print(output);
}// The main() method ends here.

// A method that takes a character as input and returns the corresponding GUI character
    static char gui_map(char in){
        char out = 'N';// N = Null/Empty
        char gm[][]={{'a','.'}
                    ,{'b','0'}
                    ,{'c','='}
                    ,{'d','+'}
                    ,{'e','1'}
                    ,{'f','2'}
                    ,{'g','3'}
                    ,{'h','-'}
                    ,{'i','4'}
                    ,{'j','5'}
                    ,{'k','6'}
                    ,{'l','X'}
                    ,{'m','7'}
                    ,{'n','8'}
                    ,{'o','9'}
                    ,{'p','/'}};

        // Checking for maps
        for(int i=0; i<gm.length; i++){
            if(gm[i][0]==in){
                out=gm[i][1];
                break;
            }
        }
        return out;
    }
}
```