Week 12 : Programming Assignment 1

Due on 2025-04-17, 23:59 IST

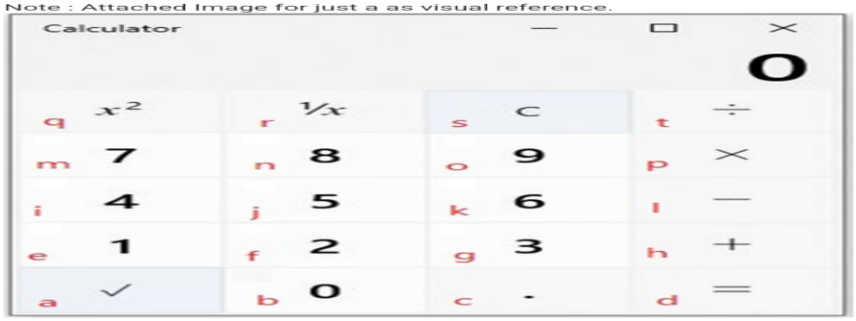
GUI-Based Advanced Calculator with Special Functions

Problem Statement

This task celebrates your learning journey — by helping you implement a very simple version of an **Advanced Calculator**, the kind that mimics buttons on a GUI.
You will be given a string of characters like `okhid` which represent button presses.
Each character corresponds to a calculator button. For example:

- `'o'` → 0, `'k'` → 6, `'h'` → +, `'d'` → =, `'q'` → Square, `'r'` → Reciprocal, etc.

Note : Attached Image for just a as visual reference.



The method `gui_map(char)` is already provided to help you convert characters into calculator buttons.
Your job is to write **only one line of code** that maps all characters from the input into GUI buttons.
This task may look long — but don't worry, we've made it super easy for you.
You've reached Week 12! That alone is a great achievement!
Let's now finish strong with something fun and satisfying

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	okhid	100.0	100.0\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 2025-04-16, 23:35 IST
Your last recorded submission was :

```
import java.util.Scanner;
public class W12_P1 {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        // Read the string representing calculator key presses
        String input = sc.nextLine();
        // Convert input string to character array
        char seq[] = input.toCharArray();
        int outflag = 0;
for(int apple=0; apple< seq.length; apple++)
        {
            seq[apple]=gui_map(seq[apple]);
            if (seq[apple]=='R' || seq[apple]=='S' || seq[apple]=='F' ||seq[apple]=='C')
                break;
        }
// The logic below performs calculation based on mapped characters
        double operand1 = 0.0,operand2 = 0.0, output = 0.0;
        String o1 = "", o2 = "";
        outerloop:
        for (int i = 0; i < seq.length; i++) {
            if (seq[i] == 'C') {
                operand1 = 0.0;
                operand2 = 0.0;
                output = 0.0;
                outflag = 0;
                break outerloop;
            } else if (seq[i] == 'R') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.sqrt(operand1);
                outflag = 1;
                break outerloop;
            } else if (seq[i] == 'S') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.pow(operand1, 2);
                outflag = 1;
                break outerloop;
            } else if (seq[i] == 'F') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.pow(operand1, -1);
                outflag = 1;
                break outerloop;
            } else {
                if (seq[i] == '+' || seq[i] == '-' || seq[i] == '/' || seq[i] == '*' || seq[i] == '=') {
                    for (int j = 0; j <= i-1; j++) o1 += seq[j];
                    operand1 = Double.parseDouble(o1);
                    for (int k = i+1; k < seq.length; k++) {
                        if (seq[k] == '+' || seq[k] == '-' || seq[k] == '/' || seq[k] == '*' || seq[k] == '=') {
                            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] == '/') output = operand1 / operand2;
                            outflag = 1;
                            break outerloop;
                        } else
                            o2 += seq[k];
                    }
                }
            }
        }
        if (outflag == 1)
            System.out.println(output);
    }
    // Function to map raw input character to calculator key
    static char gui_map(char in) {
        char out = 'N';
        char gm[][] = {
            {'a','!'}, {'b','0'}, {'c','4'}, {'d','5'}, {'e','1'}, {'f','2'},
            {'g','3'}, {'h','+'}, {'i','4'}, {'j','5'}, {'k','6'}, {'l','-'},
            {'m','7'}, {'n','8'}, {'o','9'}, {'p','*'}, {'q','S'}, {'r','F'},
            {'s','C'}, {'t','/'},
        };
        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)

```
import java.util.Scanner;
public class W12_P1 {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        // Read the string representing calculator key presses
        String input = sc.nextLine();
        // Convert input string to character array
        char seq[] = input.toCharArray();
        int outflag = 0;
for(int i = 0; i < seq.length; i++) {
    seq[i] = gui_map(seq[i]); // Convert each input character to its corresponding calculator key
}
/*
Explanation:
The array `seq` contains characters like 'o', 'k', 'h', etc.
The function `gui_map()` translates each of these into their corresponding calculator keys: '9', '6', '+', etc.
This conversion is necessary for the rest of the program to interpret and compute the result.
Final Note:
Although the solution required only a single line of code, we encourage you to read the full program carefully.
It shows how a simple character translator enables inputs to perform parsing and computes results.
Completing this problem demonstrates your ability to apply Java control structures and methods.
Keep building on your skills, and continue exploring Java beyond this course.
*/
// The logic below performs calculation based on mapped characters
        double operand1 = 0.0,operand2 = 0.0, output = 0.0;
        String o1 = "", o2 = "";
        outerloop:
        for (int i = 0; i < seq.length; i++) {
            if (seq[i] == 'C') {
                operand1 = 0.0;
                operand2 = 0.0;
                output = 0.0;
                outflag = 0;
                break outerloop;
            } else if (seq[i] == 'R') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.sqrt(operand1);
                outflag = 1;
                break outerloop;
            } else if (seq[i] == 'S') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.pow(operand1, 2);
                outflag = 1;
                break outerloop;
            } else if (seq[i] == 'F') {
                for (int j = 0; j <= i; j++) o1 += seq[j];
                operand1 = Double.parseDouble(o1);
                output = Math.pow(operand1, -1);
                outflag = 1;
                break outerloop;
            } else {
                if (seq[i] == '+' || seq[i] == '-' || seq[i] == '/' || seq[i] == '*' || seq[i] == '=') {
                    for (int j = 0; j <= i-1; j++) o1 += seq[j];
                    operand1 = Double.parseDouble(o1);
                    for (int k = i+1; k < seq.length; k++) {
                        if (seq[k] == '+' || seq[k] == '-' || seq[k] == '/' || seq[k] == '*' || seq[k] == '=') {
                            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] == '/') output = operand1 / operand2;
                            outflag = 1;
                            break outerloop;
                        } else
                            o2 += seq[k];
                    }
                }
            }
        }
        if (outflag == 1)
            System.out.println(output);
    }
    // Function to map raw input character to calculator key
    static char gui_map(char in) {
        char out = 'N';
        char gm[][] = {
            {'a','!'}, {'b','0'}, {'c','4'}, {'d','5'}, {'e','1'}, {'f','2'},
            {'g','3'}, {'h','+'}, {'i','4'}, {'j','5'}, {'k','6'}, {'l','-'},
            {'m','7'}, {'n','8'}, {'o','9'}, {'p','*'}, {'q','S'}, {'r','F'},
            {'s','C'}, {'t','/'},
        };
        for (int i = 0; i < gm.length; i++) {
            if (gm[i][0] == in) {
                out = gm[i][1];
                break;
            }
        }
        return out;
    }
}
```