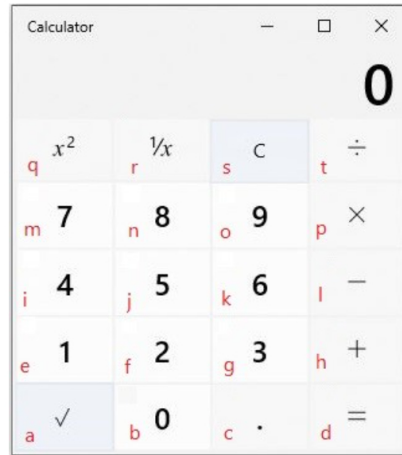


Java Week 12: Q1

Due on 2020-12-10, 23:59 IST

Complete the code to develop an extended version of the ADVANCED CALCULATOR with added special functions 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 't'.
3. You may use the already defined function **gui_map(char)**.
4. Use predefined methods from **java.lang.Math** class wherever applicable.
5. Without '=' binary operations won't give output as shown in Input_3 and Output_3 example below.
5. The calculator should be able to perform required operations on one or two operands as shown in the below example:

Input_1:
okhid

Output_1:
100.0

Input_2:
ia

Output_2:
2.0

Input_3:
okhi

Output_3:

Private Test cases used for evaluation

Test Case 1

Test Case 2

Input	Expected Output	Actual Output	Status
kq	36.0	36.0	Passed
fr	0.5	0.5	Passed

The due date for submitting this assignment has passed.

2 out of 2 tests passed.

You scored 100.0/100.

Assignment submitted on 2020-12-09, 12:38 IST

Your last recorded submission was :

```

1 import java.util.Scanner;
2 public class Question92{
3     public static void main(String args[]){
4         Scanner sc = new Scanner(System.in);
5         String input = sc.nextLine();
6
7         char seq[] = input.toCharArray();
8         int outflag=0;
9         for(int i=0; i<seq.length; i++)
10            {
11                seq[i]=gui_map(seq[i]);
12                if (seq[i]=='R' || seq[i]=='S' || seq[i]=='F' || seq[i]=='C')
13                    break;
14            }
15
16            /*Print Mapped GUI (remove comment to see the mapped sequence input)
17            for(int i=0; i<seq.length; i++){
18                System.out.print(seq[i]);
19            }
20            */
21        }
22    }

```

Course outline

How does an NPTEL online course work?

Week 0 : Assignment 0

Week 1 :

Week 2 :

Week 3 :

Week 4 :

Week 5 :

Week 6 :

Week 7 :

Week 8 :

Week 9 :

Week 10 :

Week 11 :

Week 12 :

- Lecture 56 : Case Study - I
- Lecture 57 : Case Study - II
- Lecture 58 : Case Study - III
- Lecture 59 : Case Study - IV
- Lecture 60 : Case Study - V
- Quiz: Assignment 12
- Java Week 12: Q1
- Java Week 12: Q2
- Java Week 12: Q3
- Java Week 12: Q4
- Java Week 12: Q5
- Feedback For Week 12

Solution

DOWNLOAD VIDEOS

Text Transcripts

Programming Test - (April 11 - 10AM - 12 PM)

Programming Test - (April 11 - 8PM - 10 PM)

```

22
23 // Use double type of values for entire calculation
24 double operand1=0.0;
25 String o1="";
26 double operand2=0.0;
27 String o2="";
28 double output=0.0;
29 // Perform calculation operations
30 outflag=0;
31 for(int i=0; i<seq.length; i++){
32     if(seq[i]=='C'){ //Clear
33         operand1=0.0;
34         operand2=0.0;
35         output=0.0;
36         outflag=0;
37         break outerloop;
38     }else if(seq[i]=='R'){ //Square Root
39         for(int j=0; j<i; j++){
40             o1+=Character.toString(seq[j]);
41         }
42         operand1=Double.parseDouble(o1);
43         output=Math.sqrt(operand1);
44         outflag=1;
45         break outerloop;
46     }
47     else if(seq[i]=='S'){ //Square
48         for(int j=0; j<i; j++){
49             o1+=Character.toString(seq[j]);
50         }
51         operand1=Double.parseDouble(o1);
52         output=Math.pow(operand1,2);
53         outflag=1;
54         break outerloop;
55     }else if(seq[i]=='F'){ //Inverse
56         for(int j=0; j<i; j++){
57             o1+=Character.toString(seq[j]);
58         }
59         operand1=Double.parseDouble(o1);
60         output=Math.pow(operand1,-1);
61         outflag=1;
62         break outerloop;
63     }else{
64         int r=0;
65         if(seq[i]=='+'||seq[i]=='-'||seq[i]=='/'||seq[i]=='*'||seq[i]=='='){
66             for(int j=0; j<i; j++){
67                 o1+=Character.toString(seq[j]);
68             }
69             operand1=Double.parseDouble(o1);
70             for(int k=i+1; k<seq.length; k++){
71                 if(seq[k]=='='){
72                     outflag=1;
73                     operand2=Double.parseDouble(o2);
74                     if(seq[i]=='+'){
75                         output=operand1+operand2;
76                     }else if(seq[i]=='-'){
77                         output=operand1-operand2;
78                     }else if(seq[i]=='/'){
79                         output=operand1/operand2;
80                     }else if(seq[i]=='*'){
81                         output=operand1*operand2;
82                     }
83                     break outerloop;
84                 }else{
85                     o2+=Character.toString(seq[k]);
86                 }
87             }
88         }
89     }
90 }
91 // Check if output is available and print the output
92 if(outflag==1)
93     System.out.print(output);
94 } // The main() method ends here.
95
96 // A method that takes a character as input and returns the corresponding GUI character
97 static char gui_map(char in){
98     char out = '\0'; // N = Null/Empty
99     char gm[][]={
100         {'a','R'},
101         {'b','0'},
102         {'c','.'},
103         {'d','='},
104         {'e','1'},
105         {'f','2'},
106         {'g','3'},
107         {'h','+'},
108         {'i','4'},
109         {'j','5'},
110         {'k','6'},
111         {'l','-'},
112         {'m','7'},
113         {'n','8'},
114         {'o','9'},
115         {'p','/'},
116         {'q','*'},
117         {'r','C'},
118         {'t','/'}
119     };
120     /* R = Square root
121        C = Clear/Restart
122        F = Fraction
123        S = Square
124        */
125     // Checking for maps
126     for(int i=0; i<gm.length; i++){
127         if(gm[i][0]==in){
128             out=gm[i][1];
129             break;
130         }
131     }
132     return out;
133 }
134 }
135 }
136

```

Sample solutions (Provided by instructor)

```

1 import java.util.Scanner;
2 public class Question92{
3     public static void main(String args[]){
4         Scanner sc = new Scanner(System.in);
5         String input = sc.nextLine();
6
7         char seq[] = input.toCharArray();
8         int outflag=0;
9
10        for(int i=0; i<seq.length; i++){
11            seq[i]=gui_map(seq[i]);
12            if (seq[i]=='R' || seq[i]=='S' || seq[i]=='F' || seq[i]=='C')
13                break;
14        }
15        //Print Mapped GUI (remove comment to see the mapped sequence input)
16        /*

```

```

17     for(int i=0; i<seq.length; i++){
18         System.out.print(seq[i]);
19     }
20     //
21     // Use double type of values for entire calculation
22     double operand1=0.0;
23     String o1="";
24     double operand2=0.0;
25     String o2="";
26     double output=0.0;
27     // Perform calculation operations
28     outerloop:
29     for(int i=0; i<seq.length; i++){
30         if(seq[i]=='C'){ //Clear
31             operand1=0.0;
32             operand2=0.0;
33             output=0.0;
34             outflag=0;
35             break outerloop;
36         }else if(seq[i]=='R'){ //Square Root
37             for(int j=0; j<i; j++){
38                 o1+=Character.toString(seq[j]);
39             }
40             operand1=Double.parseDouble(o1);
41             output=Math.sqrt(operand1);
42             outflag=1;
43             break outerloop;
44         }
45         else if(seq[i]=='S'){ //Square
46             for(int j=0; j<i; j++){
47                 o1+=Character.toString(seq[j]);
48             }
49             operand1=Double.parseDouble(o1);
50             output=Math.pow(operand1,2);
51             outflag=1;
52             break outerloop;
53         }else if(seq[i]=='F'){ //Inverse
54             for(int j=0; j<i; j++){
55                 o1+=Character.toString(seq[j]);
56             }
57             operand1=Double.parseDouble(o1);
58             output=Math.pow(operand1,-1);
59             outflag=1;
60             break outerloop;
61         }else{
62             int r=0;
63             if(seq[i]=='+'||seq[i]=='-'||seq[i]=='/'||seq[i]=='*'||seq[i]=='='){
64                 for(int j=0; j<i; j++){
65                     o1+=Character.toString(seq[j]);
66                 }
67                 operand1=Double.parseDouble(o1);
68                 for(int k=i+1; k<seq.length; k++){
69                     if(seq[k]=='='){
70                         outflag=1;
71                     }
72                     operand2=Double.parseDouble(o2);
73                     if(seq[i]=='+'){
74                         output=operand1+operand2;
75                     }else if(seq[i]=='-'){
76                         output=operand1-operand2;
77                     }else if(seq[i]=='/'){
78                         output=operand1/operand2;
79                     }else if(seq[i]=='*'){
80                         output=operand1*operand2;
81                     }
82                     break outerloop;
83                 }else{
84                     o2+=Character.toString(seq[k]);
85                 }
86             }
87         }
88     }
89     }
90     }
91     }
92     // Check if output is available and print the output
93     if(outflag==1)
94         System.out.print(output);
95
96     // The main() method ends here.
97
98     // A method that takes a character as input and returns the corresponding GUI character
99     static char gui_map(char in){
100         char out = '\N'; // \N = Null/Empty
101         char gm[][]={
102             {'a','R'},
103             {'b','0'},
104             {'c','.'},
105             {'d','='},
106             {'e','1'},
107             {'f','2'},
108             {'g','3'},
109             {'h','+'},
110             {'i','4'},
111             {'j','5'},
112             {'k','6'},
113             {'l','-'},
114             {'m','7'},
115             {'n','8'},
116             {'o','9'},
117             {'p','S'},
118             {'q','F'},
119             {'r','C'},
120             {'t','/'}
121         };
122         /* R = Square root
123            C = Clear/Restart
124            F = Fraction
125            S = Square
126         */
127
128         // Checking for maps
129         for(int i=0; i<gm.length; i++){
130             if(gm[i][0]==in){
131                 out=gm[i][1];
132                 break;
133             }
134         }
135         return out;
136     }
137 }
138
139

```