

FINAL PROJECT REPORT

LAB BATCH: Slot 06.09

CONTENTS:

1. Overview of project
2. Description of code
3. Ideas for future work
4. Individual contribution
5. Consolidated Diary

SciCal:

Introduction:

Often in experiments we have to separately calculate the actual value and overall error due to uncertainties in measurements of variables separately. Scical bridges his gap by simultaneously keeping a track of errors as calculations are done. The aim of our project was to design a calculator which could provide us the uncertainty of any function $f(x)$. The language would consist of simple commands using which user can provide information regarding of what function to find its uncertainty.

Project Description:

Design:

We have very systematically designed the calculator. Scical starts with a simplecpp canvas which contain all the required input buttons. Input data can be enter clicking on any desired button.

The display contains following :

1. AC
2. VAR
3. Preview
4. 1
5. 2
6. 3
7. 4
8. 5
9. 6
10. 7
11. 8
12. 9
13. 0

Simplecpp Canvas				
AC	var	prev	+	sin
1	2	3	-	cos
4	5	6	×	tan
7	8	9	/	log
=	0	un	^	{
)	ch	C	.	

14. $-$

15. $*$

16. $/$

17. $^{\wedge}$

18. $+$

19. $=$

20. un

21. $($

22. $)$

23. \sin

24. \cos

25. \tan

26. ch

27. C

28. $.$

29. pi

30. \log

Through simplecpp the buttons are made by declaring different 'Rectangle' of particular length and width and centre at particular points.

Then various text are declared at centre of respective rectangles.

Features:

The major function of this calculator is to calculate the uncertainty of almost any input function. Basic functions like sin, cos, log, power, etc are provided which can be used to create complex function.

This calculator takes input from keyboard for VARIABLE.

Control:

VAR: To create a variable whose value & uncertainty can be changed.

PREV: Used to review & edit last entered expression.

CH: Used to alter the values of predefined variables.

UN: Evaluated string as $a@b$ where a is eq(s) & b is total uncertainty.

C: Used to delete the last entered data.

DESCRIPTION OF CODE:

This project contains following functions:

`bool invalidName(string s);` returns true if it is a valid variable name

`void varCreator();` creates a new variable

`bool isDouble(string check);` checks if the string contains only a double

`void varCh();` changes the value of a particular variable

`string postfix(string c);` converts the expression to postfix

`bool isFunction(string s);` checks if the given string is the name of a function

`string to_String (double Number);` converts double to string

`double raisedTo(double i,double j);` returns i^j

`double pluss(double i,double j);` returns $i+j$

`double subs(double i,double j);` returns $i-j$

`double multi(double i,double j);` returns $i*j$

`double divid(double i,double j);` returns i/j

`double siner(double i);` returns $\sin(i)$

`double tangen(double i);` returns $\tan(i)$

double cosiner(double i);returns cos(i)

double logo(double i);returns log(i)

int numOtok(string s);returns number of tokens(whitespace separated) in s

string eq(string s);evaluates s

string prev3(string given,int j); returns previous 3 tokens

string token1(string given);returns first token in given

string token2(string given);returns second token in given

double linearun(double a,double b);returns linear uncertainty i.e. for $a \pm b$ uncertainty is $(da^2 + db^2)^{.5}$

double multiun(double a,double Dela,double b,double Delb); $d(a*b) = ((adb)^2 + (bda)^2)^{.5}$

double divun(double a,double Dela,double b,double Delb); $d(a/b) = ((Dela*Dela)/(b*b)) + ((Delb*Delb*a)/(b*b*b*b))$

double unc(string s);returns b in a@b

double value(string s);returns a in a@b

string un(string s);evaluates s as $a \pm b$ where a is eq(s) and b is total uncertainty

double dsin(double i); returns cos(i)

double dcos(double i);returns $\sin(i)$

double dtan(double i);returns $\sec^2(x)$

double dlog(double i);returns $1/x$

bool isConstant(string s);returns true if s has no uncertainty

string token2u(string given);returns 2nd token in given (slight variant of token2)

string token1u(string given); returns 2nd token in given (slight variant of token2)

double drt(double a,double Dela,double b,double Delb);returns $d(a^b)$

string prev();returns previous statement

void store(string incoming);stores previous statements

IDEAS FOR FUTURE WORK:

“Def” function may be added. It allows user to define a function say $3.14*a*b$

- i. Enter name of fn (i.e A)
- ii. No. of variables (i.e 2)
- iii. Name of variables (i.e ‘a’ <Enter> ‘b’)
- iv. Type the expression(i.e. $3.14*a*b$)
- v. Use the function in expressions (i.e. $A(a,da,b,db)$)

The values can also be linked i.e. if someone realises that he entered the he entered the wrong input then he can particularly change that value and get the required output.

The terminal window-canvas interface could be made cleaner.

We would have loved to let the user enter complete expressions through the keyboard, however this makes ‘exception handling’ very difficult & unpredictable & hence have not been covered.

INDIVIDUAL CONTRIBUTION:

Nihar Vete

Nihar wrote the algorithm of the program. Also designed the basic graphic user interface. He wrote major functions such as postfix(), var utility, eq() and un() . Debugged the program.

Rohit Bhor

Completed the functions : To check the double in string , making all basic functions , making functions to find the differentiation, finding previous line in file, various string functions, function to extract first double & second double from postfix string, etc. Contribution in completing the graphic user interface of the code. Designed the getclick() for UI with Ram. Wrote final project report.

Ram Yadav

He maintained the diary of all members of the team. He contributed in making getclick functions & UI with Rohit. He supported in making some of the functions to Nihar & Rohit. The functions include uncertainty

NOTE : All the above is based the discussion held during peer review with the consent of all team members. Please refer to Review sheet (BELOW) .

Consolidated Diary

	Discussions	Design	Programming	Testing	Documentation	Total
Nihar Vete	2 hrs	.5 hrs	11 hrs	1 hrs	.5 hrs	15 hrs
Rohit Bhor	2 hrs	1 hrs	4 hrs	1.5 hrs	3 hrs	11.5 hrs
Ram Yadav	1.5 hrs	.5 hrs	.25 hrs	2 hrs	1 hrs	4.5 hrs