1. **INTRODUCTION**

**1.1 About the Project**

**1.1.1 Scientific Calculator-An Overview**

Scientific calculator is a type of electronic calculator, usually but not always handheld, designed to calculate problems in science, engineering, and mathematics. They have almost completely replaced slide rules in traditional applications, and are widely used in both education and professional settings.

Modern scientific calculators generally have many more features than a standard four or five-function calculator, and the feature set differs between manufacturers and models; however, this project deals with 20 simple and basic mathematical functions including:

1. Addition 11. Cosec function
2. Subtraction 12. Sec function
3. Multiplication 13. Cot function
4. Division 14. X2
5. Modulus 15. root
6. Sine function 16. X!
7. Cosine function 17. 1/x
8. Tangent function 18. log
9. xy 19. ln
10. Root 20. E

**1.1.2 Objective of the PROJECT**

The main objective of the project “**SCIENTIFIC CALCULATOR**” is to help the user to do calculations with some of the main important and useful functions in MATHEMATICS. This program targets at the comfortance of the user along with proper and accurate result of the entered questions.

It helps the user to also do complex operations like Natural Logarithm, Trigonometry, Logarithm, and even the e function. Has the feature like CLEARING, EXIT and also the mathematical number PI.

1. **SYSTEM SPECIFICATION**

**2.1 Development Environment**

**2.1.1 Hardware Configuration**

PROCESSOR : Intel Core i7-5960X

RAM : HyperX Predator 3600 2X8GB

GRAPHICS ADAPTER : Nvidia GeForce GTX Titan Xp

MONITOR : AOC Agon AG352UCG

KEYBOARD : Logitech G910 Orion Spectrum

MOUSE : Logitech G903

**2.1.2 Software Configuration**

OPERATING SYSTEM : Windows 10

EDITOR : DOSBox

PROGRAMMING LANGUAGE : C++

OOP CONCEPTS USED : Data Abstraction, Data Encapsulation, Modularity, Inheritance, Polymorphism, Reusability.

FEATURES USED : FUNCTIONS

DATABASE : SQL

**2.2 ABOUT C++**

**C++ PROGRAMMING**

The C++ programming Language was developed at AT&T Bell laboratories by BjarneStroustroup in 1980s. C++ is a language developed by improving the language C. C++ was known as “C with classes”. C++ became famous and known widely by two events. (i) The formation of an ANSI (American National Standard Institute). (ii) The publication of a reference manual by Ellis and Stroustroup known as “The Annotated C++ Reference Manual”.

**C++ character set**

Character set is a set of valid characters that a language can recognize. A character represents any letter, digit or any other sign

The C++ has the following character set

Letters A-Z, a-z

Digits 0-9

Special symbols ! @ # $ % ^ & \* \_ - + = \ / , . ; ‘ ? : “ | ~

( ) { } [ ] <>

White spaces blank space, horizontal tab, enter key

Note: 1. C++ can process any of the 256 ASCII characters (American Standard Code for Information

Interchange)

2. C++ is case sensitive as it treats upper and lower case characters differently.

**Header Files**

Files ending with **.h** extension are known as header files. iostream and conio are header files. The header files have some inbuilt functions and commands in it. When we include the header files in a program, automatically these functions are also included in it.

iostream.h has the commands cout and cin included in it. conio has the functions clrscr( ) and getch ( ).

**Functions**

A function is a subprogram that acts on data and often returns a value. Every function in C++ has its code included in a pair of curly braces { }. There are two types of functions:

1. **Built-in (Library ) functions**

These functions are part of the compiler. The code is already written and saved. If we write the name of these functions in a program the whole code will be included in it.

Eg:-clrscr ( ), getch ( ).

1. **User-defined functions.**

These functions are written by the programmer according to the users requirements.

Eg:-main ( )

**TOKENS**

The smallest individual unit in a program is known as Tokens. C++ have the following tokens

* Keywords
* Identifiers
* Literals
* Punctuators
* Operators

**Keywords**

Keywords are the words that convey a special meaning to the programming language. These are reserved for special purpose and must not be used as variable names.

Egs: void, int, float, char, if, else, for, while

**Identifier**

Identifiers are the fundamental building blocks of a program. They are used as the general terminology for the names given to the different parts of the program. (Variables, functions, arrays etc)

The rules for forming the variables are given below:

1. Variables must be the sequence of letters and digits.
2. The first character of the variable must be a letter.
3. The underscore ( \_ ) is count as a letter.
4. Upper and lower case letters are different.
5. It must not be a key word.
6. It should not contain the special characters(hyphen,space,dot etc )

Valid Identifiers: - num1, \_ABC, sum, a

Invalid identifiers: - 2num, PQ\*R, float, num 3

**Literals**

Literals are data items that never change their type of value during a program run

Different kinds of literals in C++ are

* Bool literal

Data items have only two values – true or false [ 0 or 1]

* Integer constants

Data items will be of numbers without decimals

* Floating constant

Data items will have numbers with decimals

* Character constant

Data items will be of single character

* String literal

Data items will be of one or more characters

**Punctuators**

The following characters are used as punctuators in C++ (also known as separators).

[ ] ( ) { } , ; : \* \_ = #

**Operators**

Operators are tokens that trigger some computation when applied to variables and other objects in an expression.

Generally there are three type of operators

* Unary operator
* Binary operator
* Ternary operator

**UNARY OPERATOR**

These are operators that require one operator to operate upon

Eg:- + (unary plus) - (unary minus)

++ (Increment Operator) - - (Decrement operator)

**Increment Operator and Decrement operator**

A++ means A = A +1

A-- means A= A-1

**BINARY OPERATOR**

These are operators that require two operands to operate upon.

Following are some of the Binary operator

* Arithmetic Operator [ + - \* / % ]
* Relational Operator [< > <= >= == != ]
* Assignment Operator [ = /= += ]
* Logical Operator [ && || ]

**TERNARY OPERATOR**

These are operators that require three operands to operate upon. They are also known as conditional operators. [ ? :]

**Flow of control**

In a program, statements may be executed sequentially, selectively or iteratively.

**Sequence**

In this type statements are being executed sequentially. Every C++ program begins with the first statement of main ( ). Each statement in turn is executed sequentially (one after another) and ends with the last line of the main ( ).

**Selection**

In this type, the statements are executed depending upon the condition test. If a condition evaluates to true, a set of statement is followed otherwise another set of statements. The selection statements are also called decision statements as decision is taken according to the condition.

**Iteration**

In this type a set of statements will be executed repeatedly as in a loop till the exiting condition becomes true

1. **SYSTEM DESIGN**

**3.1 Functional Design**

A function is a sub program that acts on data and often returns a value. It is most important to use functions to make program handling easier as only a small part of the program is dealt at a time thereby avoiding ambiguity

Every program must contain a function main(). Program execution begins from the first line of main and ends with the last statement. Functions are the building block of C++ and the place where all activity occurs

This is a structured based program containing following functions

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **FUNCTION** | **DESCRIPTION** |
| 1 | INTRO( ) | INTRODUCTION TO THIS PROJECT |
| 2 | GRAPH( ) | GIVES GRAPHIC DESIGN TO THE PROJECT |
| 3 | DIS( ) | DISPLAY THE ITEMS AND INFORMATIONS |
| 4 | ENTER( ) | ACCEPTS THE ENTERED NUMBER AND PROCESS IT ACCORDING TO THE OPERATION ASKED |
|  | | |

**3.2 DATA FLOW DIGRAM**

**START**

**ENTER THE NUMBER**

**ENTER THE OPERAND**

**IF- 10**

**IF- 9**

**IF- 8**

**IF- 7**

**IF- 6**

**IF- 5**

**IF- 4**

**IF- 3**

**IF- 2**

**IF - 1**

N N N N N N N N N N

Y Y Y Y Y Y Y Y Y Y

Y

**IF- 20**

**IF- 16**

**IF- 15**

**IF- 14**

**IF- 13**

**IF- 12**

**IF- 11**

**IF- 171117**

**IF- 19**

**IF- 18**

N N N N N N N N N N Y

**IF- C**

**IF- P**

**IF- E**

**IF- =**

Y N N N Y

Y

**STOP**

**PRINT THE RESULT**

1. **CONCLUTION**

Making programs designed for a real-life purpose is a great way to understand programming and programming concepts. This C++ project and OOP concepts used have been of greater help to me for understanding OOP concepts and for its functioning, it can be easily modified and maintained.

This calculator is designed in a user friendly manner. This software is economically feasible. As the cost of hardware and software required is low.

The main objectives that are achieved are to do complex operations and many other functions. All the calculations are done by the computer and provide greater speed and thus reduce time consumption. It provides the user a quick response with very accurate information.

The project has been developed to meet the aspirations indicated in the modern age. An attempt has been made through this project to do all work ease and fast.

1. **BIBLIOGRAPHY**

**Textbooks:**

1. Computer Science with C++ A textbook for class XI by Sumita Arora
2. Computer Science with C++ A textbook for class XII by Sumita Arora

**Websites:**

1. http//en.wikipedia.org/
2. www.softechworld.com
3. **APPENDIX**
   1. **Source Code**

//SCIENTIFIC CALCULATOR

#include<iostream.h>

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<process.h>

#include<math.h>

#include<string.h>

#include<dos.h>

void graph();

void enter();

void dis(int left, int top);

char op2,op3;

void intro()

{

clrscr();

cout<<"\n\n\t\t\t SCIENTIFIC CALCULATOR";

cout<<"\n\t==============================================================";

cout<<"\n\n\nDISCLAIMER";

cout<<"\n----------";

cout<<"\n\nThis is a SCIENTIFIC SOFTWARE programmed to CALCULATE variables , integers and equations driven by TWENTY MATHEMATICAL OPERATIONS.";

cout<<"To operate with these twenty mathematical functions ,it is required to press ENTER KEY each time you enter a new number or function.";

cout<<" Usage of more than one operant during a sequence of operations is prohibited.In order to get the result, '=' must be used as operand";

cout<<"Now 'SIT BACK AND ENJOY.....'";

cout<<"\n\n\n\n\n\n\t\t\t PRESS ENTER TO PROCEED.....";

delay(1000);

getch();

}

void graph()

{

int gd=DETECT,gm,left=200,top=150,right=400,bottom=200;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

cout<<"\n\n\n\n================================================================================";

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n================================================================================";

rectangle(left,top-40,right+50,bottom+180);

outtextxy(left+45,top-30,"SCIENTIFIC CALCULATOR");

rectangle(left+10,top-10,right+40,bottom-20);

gotoxy(28,11);

cout<<"0";

outtextxy(left+5,top+45," C");

rectangle(left+10,top+60,right+40,bottom+150);

dis(left,top);

enter();

getch();

closegraph();

}

void dis(int left, int top)

{

outtextxy(left+15,top+75," ");

outtextxy(left+15,top+85,"1.+ 6.Sin 11.Asin 16.X!");

outtextxy(left+15,top+95," ");

outtextxy(left+15,top+105,"2.- 7.Cos 12.Acos 17.1/X");

outtextxy(left+15,top+115," ");

outtextxy(left+15,top+125,"3.\* 8.Tan 13.Atan 18.log");

outtextxy(left+15,top+135," ");

outtextxy(left+15,top+145,"4./ 9.X^y 14.X^2 19.ln");

outtextxy(left+15,top+155," ");

outtextxy(left+15,top+165,"5.% 10.rtXy 15.sqrt 20.e-");

outtextxy(left+5,top+210," PI: P");

outtextxy(left+180,top+210,"EXIT: E");

}

void enter()

{

int i=0,s=0,d=0,p=0,q=0,m=0,si=0,co=0,ta=0,po=0,power1=0,as=0,ac=0,at=0,sq2=0,fa=0,nu=0,ln=0,lo=0,en=0,pine=0;

int p2=0;

int MS,MR,mod[5],a,b;

char op[5];

float An[20],sum,diff,pro,quo,e,pi;

double sine,cosine,tangent,asine,acosine,atangent,ans,powe,power2,power,squart,f,sw;

gotoxy(28,11);

cin>>An[0];

do

{

gotoxy(28,11);

cout<<"Enter the operant ";

gets(op);

gotoxy(28,11);

cout<<" ";

if(strcmp(op,"c")==0)

graph();

else if(strcmp(op,"e")==0)

exit(0);

else if(strcmp(op,"p")==0)

{

pine=1;

pi=3.14;

gotoxy(28,11);

cout<<" ";

ans=pi\*An[0];

break;

}

else if(strcmp(op,"1")==0)

{ if(s==0)

sum=An[0];

s=s+1;

i=i+1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[i];

sum=sum+An[i];

continue;

}

else if(strcmp(op,"2")==0)

{ if(d==0)

diff=An[0];

d=d+1;

i=i+1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[i];

diff=diff-An[i];

continue;

}

else if(strcmp(op,"3")==0)

{ if(p==0)

pro=An[0];

p=p+1;

i=i+1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[i];

pro=pro\*An[i];

continue;

}

else if(strcmp(op,"4")==0)

{ if(q==0)

quo=An[0];

q=q+1;

i=i+1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[i];

quo=quo/An[i];

continue;

}

else if(strcmp(op,"5")==0)

{ if(m==0)

mod[0]=An[0];

m=1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>mod[1];

mod[2]=mod[0]%mod[1];

break;

}

else if(strcmp(op,"6")==0)

{ si=1;

sine=(An[0]\*3.14)/180;

ans=sin(sine);

break;

}

else if(strcmp(op,"7")==0)

{ co=1;

cosine=(An[0]\*3.14)/180;

ans=cos(cosine);

break;

}

else if(strcmp(op,"8")==0)

{ ta=1;

tangent=(An[0]\*3.14)/180;

ans=tan(tangent);

break;

}

else if(strcmp(op,"9")==0)

{

po=po+1;

a=An[0];

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[1];

b=An[1];

power=pow(a,b);

break;

}

else if(strcmp(op,"10")==0)

{ power1=power1+1;

f=An[0];

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cin>>An[1];

sw=An[1];

powe=pow(f,1/sw);

break;

}

else if(strcmp(op,"11")==0)

{

as=1;

asine=(An[0]\*3.14)/180;

ans=sin(1/asine);

break;

}

else if(strcmp(op,"12")==0)

{

ac=1;

acosine=(An[0]\*3.14)/180;

ans=acos(acosine);

break;

}

else if(strcmp(op,"13")==0)

{

at=1;

atangent=(An[0]\*3.14)/180;

ans=atan(atangent);

break;

}

else if(strcmp(op,"14")==0)

{

p2=1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

power2=pow(An[0],2);

break;

}

else if(strcmp(op,"15")==0)

{

sq2=1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

squart=sqrt(An[0]);

break;

}

else if(strcmp(op,"16")==0)

{

fa=1;

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

for(i=1;i<=An[0];i++)

ans\*=i;

break;

}

else if(strcmp(op,"17")==0)

{

nu=1;

gotoxy(28,11);

cout<<" ";

ans=1/An[0];

break;

}

else if(strcmp(op,"19")==0)

{

ln=1;

gotoxy(28,11);

cout<<" ";

ans=log(An[0]);

break;

}

else if(strcmp(op,"18")==0)

{

lo=1;

gotoxy(28,11);

cout<<" ";

ans=log10(An[0]);

break;

}

else if(strcmp(op,"20")==0)

{

en=1;

gotoxy(28,11);

cout<<" ";

e=2.718;

ans=An[0]\*e;

break;

}

}while(strcmp(op,"=")!=0);

if(s>=1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<sum;

}

else if(d>=1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<diff;

}

else if(p>=1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<pro;

}

else if(q>=1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<quo;

}

else if(m==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<mod[2];

}

else if(si==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(co==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(ta==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(po==1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<power;

}

else if(power1==1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<powe;

}

else if(as==1)

{ gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(ac==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(at==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(p2==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<power2;

}

else if(sq2==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<squart;

}

else if(nu==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(fa==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(ln==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(lo==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(en==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else if(pine==1)

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<ans;

}

else

{

gotoxy(28,11);

cout<<" ";

gotoxy(28,11);

cout<<"error";

}

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\tDO YOU WANT TO CONTINUE ? ";

cin>>op2;

if(op2=='y'||op2=='Y')

graph();

else

{

gotoxy(28,11);

cout<<"Exit ? ";

cin>>op3;

if(op3=='y'||op3=='Y')

exit(0);

else

graph();

}

}

void main()

{ clrscr();

intro();

graph();

getch();

}

**7.2 Output Screen Layouts**





























































