VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

"A MINI PROJECT REPORT"
(Subject Code:18CSMP68)
ON
" Scientific Calculator "

Submitted in partial fulfillment for the requirements for the Award of Degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
BY

Karthik R (1EP20CS037)
Mohnish Reddy G (1EP20CS055)
Mukeshreddy Nagireddygari (1EP20CS056)

UNDER THE GUIDANCE OF

Dr. Heena Kousar
Associate Professor
Dept. of CSE, EPCET

Mrs. Shammi L
Assistant Professor
Dept. of CSE, EPCET

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering
EAST POINT COLLEGE OF ENGINEERING AND TECHNOLOGY
Bidarahalli, Bengaluru – 560 049

2022-2023

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering

Approved by AICTE New Delhi | Affiliated to VTU, Belagavi,
Virgo Nagar, Bengaluru-560049

CERTIFICATE

This is to certify that **Karthik R, Mohnish Reddy G and Mukeshreddy Nagireddygari** bearing USN **1EP20CS037, 1EP20CS055 and 1EP20CS056** respectively are bonafide students of **East Point College of Engineering and Technology, Bengaluru,** has successfully completed MAD Mini Project entitled " **Scientific Calculator** " in partial fulfillment of the requirements for 6th Semester **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2022 - 2023. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report. The mini project report has been approvedas it satisfies the academic requirements as part of the course **Mobile Application and Development laboratory** with mini project **18CSMP68** prescribed for the said degree.

Signature of the Guide1	Signature of the Guide2	Signature of the HoD
Dr. Heena Kousar	Mrs. Shammi L	Dr. I Manimozhi
Associate Professor	Assistant Professor	HOD
Dept. of CSE, EPCET	Dept. of CSE, EPCET	Dept. of CSE, EPCET

Name of the Examiners	Signature with Date
1._____	_____
2._____	_____

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. We would like to take this opportunity to thank them all.

First and foremost, we would like to express our sincere regards and thanks to **Mr. Pramod Gowda** and **Mr. Rajiv Gowda,** CEOs East Point Group of Institutions, Bangalore, for providing necessary infrastructure and creating good environment.

We express our gratitude to **Dr. Yogesh G S,** Principal, EPCET who has always been a great source of inspiration.

We express our sincere regards and thanks to **Dr. I. Manimozhi (I/C), Head of the Department**, Department of Computer Science and Engineering, EPCET, Bangalore, for his encouragement and support.

We are grateful to acknowledge the guidance and encouragement given to us by **Dr. Heena Kousar, Associate Professor and Mrs. Shammi L**, **Assistant Professor**, Department of Computer Science and Engineering, EPCET, Bangalore, who has rendered a valuable assistance.

We also extend our thanks to all the faculties of the **Department of Computer Science and Engineering, EPCET**, Bangalore, who have encouraged us throughout the course of the Mini Project Work.

Last, but not the least, we would like to thank our family and friends for their inputs to improve the Mimi Project works.

KARTHIK R [1EP20CS037]

MOHNISH REDDY G [1EP20CS055]

MUKESHREDDY NAGIREDDYGARI [1EP20CS056]

ABSTRACT

Android is the most widely used mobile platform, and unfortunately, it is widely targeted mobile platform as well. The framework concentrates on developing an app with malicious behaviour which helps us in understanding the functionality of any malicious app. The intent of the malicious activity is to automatically answer calls from specific phone number. Once the malicious application is installed, a call can be made to the infected device for eavesdropping on the conversation taking place around the infected device. This helps in understanding the functionality of telephony manager. Calculators are part and parcel of modern education. Involvement of science and engineering in different fields of knowledge is increasing with each bit of time is passed by, and they are playing a role in description and characterization of the delicate phenomena of nature arising day by day. These fields of knowledge and mathematics in particular, are influencing even those distant branches of knowledge, which were so far imagined to be free of mathematics. Even art is not free of mathematics and there exists mathematical art computations are getting lengthy and complex specially in design and analysis of engineering systems.

Scientific calculators are handy tools. But an efficient computation is a skill that can be developed.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Calculator is used to our calculation easier. But this android based calculator project will not do only simple calculations but more than that. This new android calculator will store your mathematical operations and your answers in their memory by which you can check your results and operations which you have performed earlier. To make complex task easier, some extra features has been added in this android calculator and these are like unit conversion, fractional calculations, trigonometric calculations and unit conversion also. This calculator will not only work in single mode, but different versions available under one hood like: users can switch to scientific type or to engineering type and much more. To develop antimalware or to protect the device from being affected, it is important to understand the functioning of malicious application. To do so, it is important to understand how a malicious application can access unwanted permissions and how a malicious application can affect the user (financial loss, data loss, etc).

This is a basic project for beginners, the android calculator app will help us to do various arithmetic calculations. This android application will have a user interface with numbers and arithmetic operations.

For the development of this application, we will make use of Android Studio.

SCOPE

The main goal of the developed system is to understand the functioning of malicious apps and to understand how a malicious application accesses telephony methods to answer the call automatically and listen to the conversation taking place 10 around the infected device. Most of the malicious applications use permissions or API calls related to android telephony framework to incur financial losses or to access sensitive information. In the developed system, malicious code is embedded in normal application. When the unsuspecting user installs this application, malicious code is executed as well, which accesses telephony methods to answer the call automatically, thereby allowing hackers to listen to the conversation taking place around the infected device.

Existing System:

All we know that, while using simple calculator, we don't able to get all these features at one place and mostly rely on manual work. Because of this extensive manual work, there is some delay and if in rush condition then not able to get the correct results. There sometimes for the purpose of numerical and statistical numerical, we need to make conversion of results into octal or hexadecimal format, which again not provided through normal or scientific calculator.

Proposed System:

It's the android calculator which will remember your all input, into its memory and separate all your digits with commas automatically, so that it can be easily identified. Calculation is sometime boring, but its interactive look and versatile feature will provide you an exciting calculation environment. Because of touchy nature of your nature, using its smart editor, you don't have to write everything. If there's an error, you can click on location and rectify it. It will provide all the basic operations in normal mode, but users can also choose the scientific mode or fixed mode display for their use.

1.2. Importance of the project:

1. Fractional calculations can be easily performed using this android calculator.
2. Users will able to check their previous results in history sections and also able to send it to memory for further process.
3. Grouping of digits and decimal points will be handle by this android calculator.
4. Users will able to enter digits and operators which will also be displayed over the input panel.
5. Users can make conversions by using conversion unit section to make their calculation on hand.
6. Power source (mains electricity, battery and/or solar cell)
7. Keypad (input device) – consists of keys used to input numbers and function commands (addition, multiplication, square-root, etc.)
8. Display panel (output device) – displays input numbers, commands and results.

CHAPTER 2

SYSTEM DESIGN

In order to embed the malicious code, a normal application which in this case is a simple calculator app is developed. The malicious code is then embedded within the calculator app. To answer a call, user permissions are defined and a receiver is created which accepts broadcast events. One of the instances of hidden class of android telephony framework is accessed, which controls useful methods of call state. The call is answered automatically if and only if it is received from one particular number, that number is defined and compared with the incoming call number. If both match the call is accepted automatically, otherwise the app will behave normally. When the user clicks the power button the call is automatically disconnected and the call log for the number is deleted accordingly.

2.1 Design tasks and Requirements

Design a calculator based on Android system to realize the arithmetic of addition, subtraction, multiplication and division, and the operation of clearing and cancelling. The interface design should be as simple and beautiful as possible, with good interaction, and the program should have good robustness.Display the operation result in another line.

In the whole process of input and calculation, first of all, the input data should be processed with fault tolerance, which is conducive to improving the user experience. For example, in the process of input, operators can't input continuously, and zero can't appear at the beginning of integer data. In the process of programming, we need to design data structure to store numbers, operators and fault tolerant tags.

For continuous input of multiple groups of data, such as 3+8/2-98 or (1+2) 3 +5-4/2, it should be able to distinguish the numbers and operators, and operate according to the priority.

2.2. System Architecture of Scientific Calculator Application

Fig.2.1 shows the design architecture of the Scientific Calculator application. This application is an example of Scientific Calculator and Factorial Function, BMI operation system designed

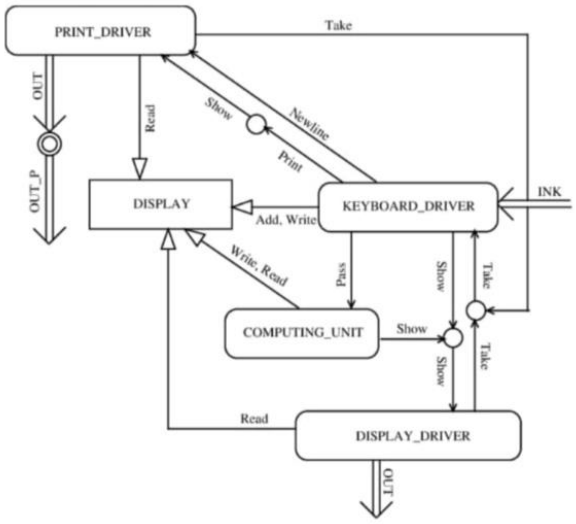

Fig.2.1: Scientific Calculator Application Architecture

to serve some mathematical functional needs. NWP refers to the simulation and prediction of the atmosphere with a computer model, and WRF is a set of software for this.

2.3 Design Principle and Structured Diagram

The expression string is processed, the numbers and operators are stored in two stacks respectively, and the design method is used to process the two stacks to get the corresponding calculation results. The operator design priority, multiplication and division priority is set to 2, and the addition and subtraction priority is set.

Pop up two numbers from the number stack, calculate the operation results of these two numbers under the operator symbol 1, press the results into the number stack, press the symbol 2 into the operator stack, and finally return one.

Pop up three numbers from the number stack, calculate the operation results of the last two numbers under the operator symbol 2, press the operation results into the stack after the end, press the remaining first number into the stack, press the operator symbol 1 into the stack, and finally return one.

2.4 Detailed Design

The input module mainly describes the calculator keyboard and keyboard monitoring, that is,

it is mainly responsible for reading the user's keyboard input and responding to the touch-screen keys. When the user clicks the key or the screen, the listener will call the corresponding function keys, which are reset, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, , , , @, left bracket, right bracket, etc. The display module mainly describes the display area of the calculator, which is used to display the data input by the user, the final calculation results and some other information.

According to the different input polynomials (such as input 3+8/2-9 * 8), it is necessary to divide the polynomials, find out the numbers and symbols and save them respectively, and then calculate them according to the priority of operators.

2.5 Dataflow Diagram

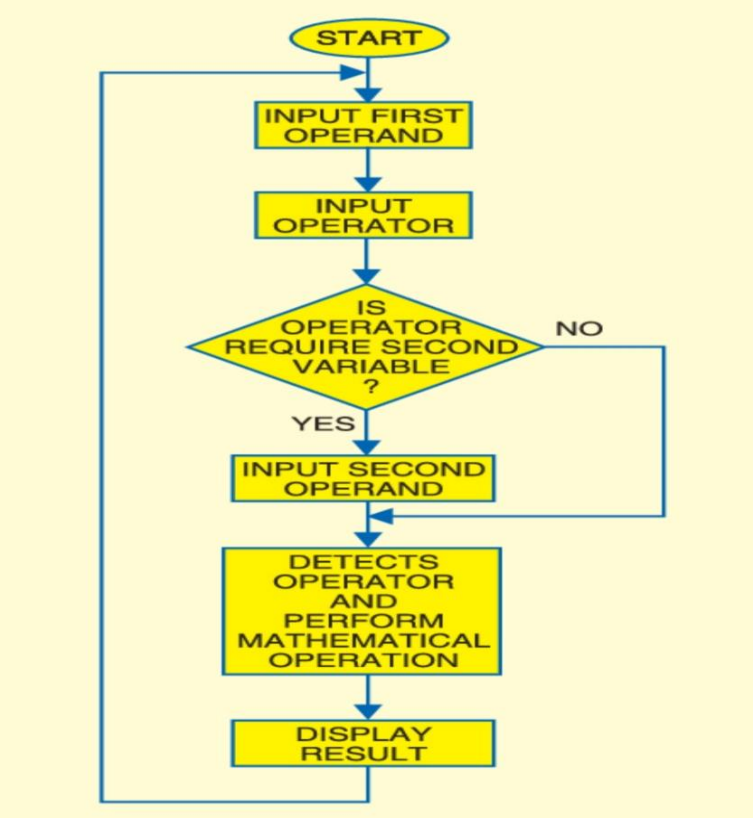

Fig 2.2: Dataflow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

CHAPTER 3

SYSTEM REQUIREMENT AND SPECIFICATION

3.1. Hardware Requirements

- 64-bit Microsoft® Windows® 8/10
- 8 GB RAM or more
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution.

3.2. Software Requirements

Android studio version 4:

- The AOSP master branch is traditionally developed and tested on Ubuntu Long Term Support (LTS) releases.
- But other distributions may be used. See Establishing a Build Environment for additional required packages and the commands to install them.
- Your workstation must have the software listed below.
- These requirements apply to the AOSP master branch. For Android versions 8.0.
    - (Oreo or O) through
    - 5.0 (Lollipop or L)
- Consider using the included Docker file to ease installation of all required packages.
- For the manual method, see Supporting Older Versions.

CHAPTER 4

IMPLEMENTATION

A Scientific Calculator is useful for situations where we need to calculate some complex things like logs or trigonometry. In such cases, the normal calculator won't be useful for us. So therefore, we are here to develop a Scientific Calculator.

This calculator will have the following:

- First, it will have a screen to display the user inputs and by default, it will display a "0", To make this we will use Text View.
- Then there will be another screen that shows the result and the operator. It will be a Text View too.
- Then we'll have the keys, that will have numbers as well as the operators on it. We'll use Buttons for this.
- We'll use a Linear layout for this so that we can arrange the buttons in the desired manner.
- **Resource folder:** This folder will have all the resources that are useful for this project.

Following are the list of resource files that are used in our project

- **Drawable:** It has the layout and designing of the application components.
- **Colors:** It has all the colors that are used in the project.
- **Style:** Here the styling of the text or components is done.
- **String:** This file defines all the strings that are necessary for the project.

2.1 SOURCE CODE

```
package com.example.myscientificcalci;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    Button
b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdot,bpi,bequal,bplus,bmin,bmul,bdiv,binv,bsqrt,bsquare,bfac
t,bln,blog,btan,bcos,bsin,bb1,bb2,bc,bac;
    TextView tvmain,tvsec;
    String pi = "3.14159265";
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1 = findViewById(R.id.b1);
        b2 = findViewById(R.id.b2);
        b3 = findViewById(R.id.b3);
        b4 = findViewById(R.id.b4);
        b5 = findViewById(R.id.b5);
        b6 = findViewById(R.id.b6);
        b7 = findViewById(R.id.b7);
        b8 = findViewById(R.id.b8);
        b9 = findViewById(R.id.b9);
        b0 = findViewById(R.id.b0);
        bpi = findViewById(R.id.bpi);
        bdot = findViewById(R.id.bdot);
        bequal = findViewById(R.id.bequal);
```

```
bmin = findViewById(R.id.bmin);
bmul = findViewById(R.id.bmul);
bdiv = findViewById(R.id.bdiv);
binv = findViewById(R.id.binv);
bsqrt = findViewById(R.id.bsqrt);
bsquare = findViewById(R.id.bsquare);
bfact = findViewById(R.id.bfact);
bln = findViewById(R.id.bln);
blog = findViewById(R.id.blog);
btan = findViewById(R.id.btan);
bsin = findViewById(R.id.bsin);
bcos = findViewById(R.id.bcos);
bb1 = findViewById(R.id.bb1);
bb2 = findViewById(R.id.bb2);
bc = findViewById(R.id.bc);
bac = findViewById(R.id.bac);
tvmain = findViewById(R.id.tvmain);
tvsec = findViewById(R.id.tvsec);
b1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      tvmain.setText(tvmain.getText()+"1");
   }
});
b2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      tvmain.setText(tvmain.getText()+"2");
   }
});
b3.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      tvmain.setText(tvmain.getText()+"3");
```

```
}   });
        b4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvmain.setText(tvmain.getText()+"4");
            }
        });
        b5.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvmain.setText(tvmain.getText()+"5");
            }
        });
        b6.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvmain.setText(tvmain.getText()+"6");
            }
        });
        b7.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvmain.setText(tvmain.getText()+"7");
            }
        });
        b8.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvmain.setText(tvmain.getText()+"8");
            }
        });
        b9.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
        tvmain.setText(tvmain.getText()+"9");           }
    });
    b0.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"0");
        }
    });
    bdot.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+".");
        }
    });
    bac.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText("");
            tvsec.setText("");
        }
    });
    bc.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String val = tvmain.getText().toString();
            val = val.substring(0, val.length() - 1);
            tvmain.setText(val);
        }
    });
    bplus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"+");
        }
```

```
});       bmin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvmain.setText(tvmain.getText()+"-");
    }
});
bmul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvmain.setText(tvmain.getText()+"×");
    }
});
bdiv.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvmain.setText(tvmain.getText()+"÷");
    }
});
bsqrt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String val = tvmain.getText().toString();
        double r = Math.sqrt(Double.parseDouble(val));
        tvmain.setText(String.valueOf(r));
    }
});
bb1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvmain.setText(tvmain.getText()+"(");
    }
});
bb2.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {

    tvmain.setText(tvmain.getText()+")");
        }
    });
    bpi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvsec.setText(bpi.getText());
            tvmain.setText(tvmain.getText()+pi);
        }
    });
    bsin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"sin");
        }
    });
    bcos.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"cos");
        }
    });
    btan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"tan");
        }
    });
    binv.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```
            tvmain.setText(tvmain.getText()+"^"+"(-1)");
        }
    });
    bfact.setOnClickListener(new View.OnClickListener() {

    @Override
        public void onClick(View v) {
            int val = Integer.parseInt(tvmain.getText().toString());
            int fact = factorial(val);
            tvmain.setText(String.valueOf(fact));
            tvsec.setText(val+"!");
        }
    });
    bsquare.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            double d = Double.parseDouble(tvmain.getText().toString());
            double square = d*d;
            tvmain.setText(String.valueOf(square));
            tvsec.setText(d+"²");
        }
    });
    bln.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"ln");
        }
    });
    blog.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tvmain.setText(tvmain.getText()+"log");
        }
    });
```

```
    bequal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String val = tvmain.getText().toString();
            String replacedstr = val.replace('÷','/').replace('×','*');
            double result = eval(replacedstr);

           tvmain.setText(String.valueOf(result));
            tvsec.setText(val);
        }
    });
}
int factorial(int n)
{
    return (n==1 || n==0) ? 1 : n*factorial(n-1);
}
public static double eval(final String str) {
    return new Object() {
        int pos = -1, ch;
        void nextChar() {
            ch = (++pos < str.length()) ? str.charAt(pos) : -1;
        }
        boolean eat(int charToEat) {
            while (ch == ' ') nextChar();
            if (ch == charToEat) {
                nextChar();
                return true;
            }
            return false;
        }
        double parse() {
            nextChar();
            double x = parseExpression();
            if (pos < str.length()) throw new RuntimeException("Unexpected: " + (char)ch)
```

```
    return x;
}
    double parseExpression() {
    double x = parseTerm();
    for (;;) {
        if      (eat('+')) x += parseTerm(); // addition
        else if (eat('-')) x -= parseTerm(); // subtraction
        else return x;
    }
}
double parseTerm() {
    double x = parseFactor();
    for (;;) {
        if      (eat('*')) x *= parseFactor(); // multiplication
        else if (eat('/')) x /= parseFactor(); // division
        else return x;
    }
}
double parseFactor() {
    if (eat('+')) return parseFactor(); // unary plus
    if (eat('-')) return -parseFactor(); // unary minus
    double x;
    int startPos = this.pos;
    if (eat('(')) { // parentheses
        x = parseExpression();
        eat(')');
    }else if((ch>='0'&&ch<='9')||ch==',')

  {
        while ((ch >= '0' && ch <= '9') || ch == '.') nextChar();
        x = Double.parseDouble(str.substring(startPos, this.pos));
    } else if (ch >= 'a' && ch <= 'z') { // functions
        while (ch >= 'a' && ch <= 'z') nextChar();
        String func = str.substring(startPos, this.pos);
```

```
            x = parseFactor();
            if (func.equals("sqrt")) x = Math.sqrt(x);
            else if (func.equals("sin")) x =Math.sin(Math.toRadians(x));
            else if (func.equals("cos")) x =Math.cos(Math.toRadians(x));
            else if (func.equals("tan")) x =Math.tan(Math.toRadians(x));
            else if (func.equals("log")) x = Math.log10(x);
            else if (func.equals("ln")) x = Math.log(x);
            else throw new RuntimeException("Unknown function: " +func);
          } else {
            throw new RuntimeException("Unexpected: " + (char)ch);
          }
          if (eat('^')) x = Math.pow(x, parseFactor()); // exponentiation
          return x;
        }
      }.parse();
    }

}
```

CHAPTER 5

SYSTEM TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

5.1 Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes cntering data and ascertaining if the value matches to the type and size supported by python. The various controls are tested to ensure that each performs its action as required.

5.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

5.3 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

5.4 Test case

Test no	Test name	Input	Actual output	Expected output	Status
1	Addition	7,2	9	9	Pass
2	Subtraction	8,5	3	3	Pass
3	Multiplication	9,5	45	45	Pass
4	Division	9,3	3	3	Pass

Table 5.1 Test Cases

CHAPTER 6

SNAPSHOTS

Scientific calculators are used widely in situations that require quick access to certain mathematical functions, especially those that were once looked up in mathematical tables, such as trigonometric functions or logarithms. They are also used for calculations of very large or very small numbers, as in some aspects of astronomy, physics, and chemistry. They are very often required for math classes from the junior high school level through college, and are generally either permitted or required on many standardized tests covering math and science subjects; as a result, many are sold into educational markets to cover this demand, and some high-end models include features making it easier to translate a problem on a textbook page into calculator input, e.g. by providing a method to enter an entire problem in as it is written on the page using simple formatting tools.

ADDITION OPERATION:

Fig 6.1 Addition operation

SUBSTRACTION:

Fig 6.2 Substraction operation

MULTIPLICATION OPERATION:

Fig 6.3 Multiplication operation

DIVISION OPERATION:

Fig 6.4 Division operation

TRIGNOMETRIC FUNCTION:

Fig 6.5 Trignometric function

ARITHEMATIC EXPRESSION:

Fig 6.6 Arithematic expression

CONCLUSION

In conclusion, the scientific calculator project has been successfully completed, meeting the desired objectives and requirements. This project aimed to design and develop a versatile calculator capable of performing a wide range of complex mathematical calculations and functions.

Throughout the project, a systematic approach was followed, starting with extensive research on scientific calculator functionalities and user requirements. This research helped in identifying the key features and operations that needed to be implemented in the calculator.

The development process involved designing the user interface, implementing the mathematical algorithms, and integrating various functions such as arithmetic operations, trigonometric functions, logarithmic functions, and more. The calculator was also equipped with memory functions, parentheses support, and error handling mechanisms to ensure accurate and reliable calculations.

The final product demonstrates a user-friendly interface with intuitive button layouts and a clear display. The calculator has been thoroughly tested to ensure accurate results and error-free operation in various scenarios. Different test cases were executed to verify the correctness of calculations and validate the implemented features.

The project has not only provided a functional scientific calculator but has also enhanced my understanding of software development methodologies, algorithms, and mathematical concepts. It has given me valuable insights into the complexities involved in building a robust calculator and the importance of thorough testing.

Overall, the scientific calculator project has been a rewarding experience, resulting in a successful implementation of a versatile calculator with a comprehensive set of functions. The calculator meets the project objectives and provides users with a reliable tool for their mathematical calculations. Further improvements and enhancements can be made in the future to expand its functionality and usability.

REFERENCES

[1]www.google.co.in

[2] Various videos from www.youtube.com

[3] For Flow Chart of Scientific Calculator : https://www.researchgate.net/figure/Flow-chart-of-working-of-scientific-calculator_fig1_303364866

[4]https://en.wikipedia.org/wiki

[5] Books:- 1. Android Programming: The Big Nerd Ranch Guide 2012-Bill Phillips

2. Head First Android Development- David Griffiths