

SRM UNIVERSITY DELHI-NCR SONEPAT

Project Report
on
"CALCULATOR"



for the partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology.

Batch 2018-2022

Submitted By:-

VANSH AGRAWAL

Reg.No :-10618210017

B.Tech.(CSE-E)1stYear

BIG DATA AND ANALYTICS

Certificate

This is to certify that the project titled "CALCULATOR" is a Bonafede work carried by Mr. VANSI AGRAWAL, Reg. no 10618210017 in the partial fulfillment for the award of Degree of "**Bachelor of Technology.**" Through SRM UNIVERSITY, DELHI - NCR SONEPAT, HARYANA

Internal Teacher

{Ms. GARIMA VIJ}

Head of Department

{Department Of computer Science}

ACKNOWLEDEMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extends their wholehearted co-operation and have helped in completing this project successfully .

First of all, I would like to thanks **Ms. Ajay Sharma** for creating opportunities to take undertake me in the esteemed organization.

i m highly indebted and graceful to their strict supervision ,constant encouragement inspiration and guidance ,which ensure the worthiness of my work working under them was an enrich experience.

I would like to thank to **Ms.Garima** my internal project guide who guide me alot in completing this project. I would also like to thank my parents & project mate for guiding and encouraging me throughout the duration of project.

In all I found a congenial work environment in "IBM" and this completion of project will mark a new beginning for me in the coming days.

VANSH AGRAWAL
(Signature of Student)

DECLARATION-cum-UNDERTAKING

I hereby declare that the work which is being presented in project Report titled "**Calculator**" in the partial fulfillment for the award of Degree of "**Bachelor** of Technology." and submitted to the department of SRM UNIVERSITY Sonipat is an authentic record and is carried out by me during the period January-June2019, under the guidance and supervision of Mr.Ajay Sharma , Department of computer science and Ms.Garima Vij (Trainer) of Project.

I hereby undertake that the matter embodied in this project report is my own work and has not submitted by me or any other candidate for the award of any degree anywhere else.

Signature of Candidate

Signature of Supervisor

INTRODUCTION

Software application development began with [desktop applications](#), which could be used on standalone machines only. Word processors and media players can be considered to be typical desktop applications. By definition, a desktop application means any software that can be installed on a single computer (laptop or a desktop) and used to perform specific tasks. Some desktop applications can also be used by multiple users in a networked environment.

SYSTEM REQUIREMENTS

- JAVA
- DB2
- NET BEANS

LITERATURE REVIEW

INTRODUCTION TO LANGUAGE USED TO CREATE A DESKTOP APPLICATION

JAVA

- **Java** is a general computer, high level, robust, secured and object-oriented programming language developed by Sun Microsystems. Originally called "Oak", by its inventor James Gosling.
- **Java** was designed with several innovative features. These include a language that is entirely object oriented, and the ability to write an application once and move it to virtually any platform. Java is a **programming language** and a **platform**.

- **Platform:** Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

CREATION OF JAVA

- Java was developed by James Gosling, Patrick Naughton, Mike Sheridan at Sun Microsystems Inc. in 1991. It took 18 months to develop the first working version. The initial name was Oak but was renamed to java in 1995. James Goslings initiated the Java language project in June 1992 for use in one of his many set-top box projects. The language, initially called Oak after an oak tree that stood outside Gosling's office, also went by the name Green and ended up later being renamed as Java from a list of random words.
- Sun released the first public implementation as Java 1.0 in 1995. It promised **Write Once, Run Anywhere** (WORA), providing no-cost run-times on popular platforms.
- On 13th November 2006, Sun released much of Java as free and open source software under the terms of the GNU General Public License (GPL).
- On 8th May 2007, Sun finished the process, making all of Java's core code free and open-source, aside from a small portion of code to which Sun did not hold the copyright.

WHERE IT IS USED?

According to Sun, 3 billion devices run java. There are many devices where java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in, javatpoint.com etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games etc

TYPES OF JAVA APPLICATION

There are mainly 4 type of applications that can be created using java programming:

1) Standalone Application

It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

2) Web Application

An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.

3) Enterprise Application

An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

4) Mobile Application

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications

FEATURES OF JAVA

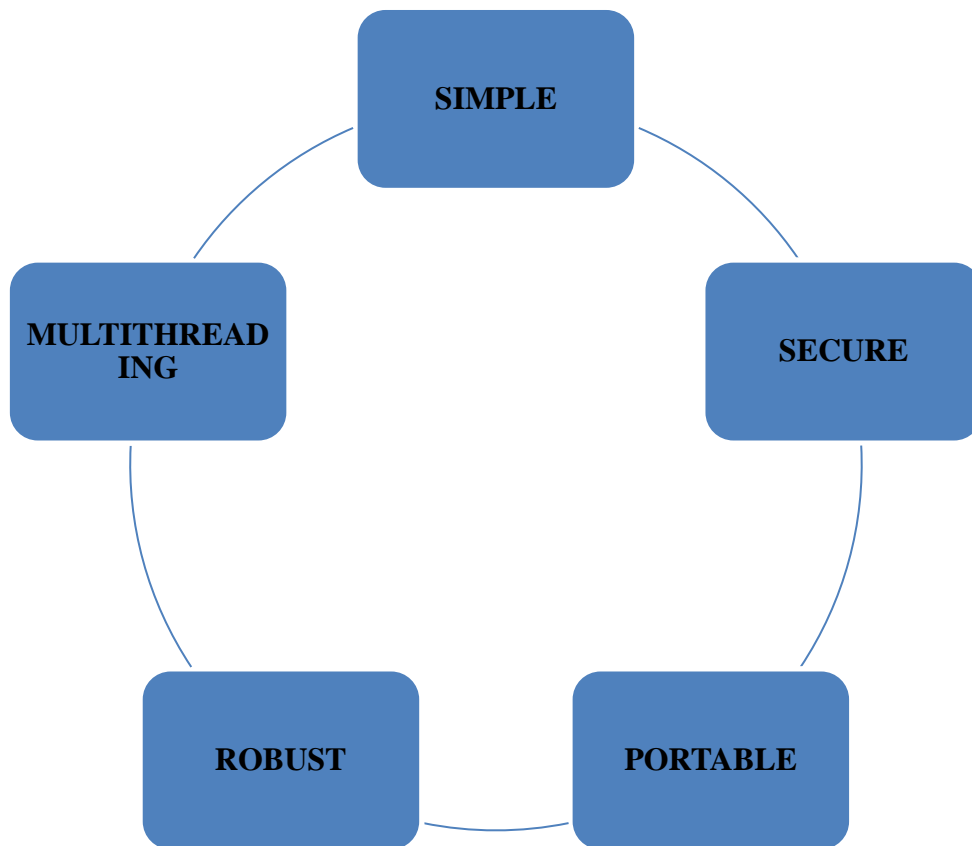


FIGURE 1: FEATURES OF JAVA

There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

1. Simple
2. Object-Oriented
3. Platform independent
4. Secured
5. Robust
6. Portable
7. High Performance

8. Multithreaded

- **Simple**

Java is easy to learn and its syntax is quite simple, clean, and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Example: Pointers and Operator Overloading are not there in java but were an important part of C++.

- **Object Oriented**

In java everything is Object which has some data and behavior. Java can be easily extended as it is based on Object Model.

Basic concepts of OOPs are:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

- **Robust**

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception and Exception Handling.

- **Platform Independent**

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provides security. Any machine with Java Runtime Environment can run Java Programs.

- **Secure**

When it comes to security, Java is always the first choice. With java secure features it is enabled us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system. OS, hence it is more secure.

- **Multi-Threading**

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications etc.

- **Portable**

We may carry the java bytecode to any platform.

- **High-performance**

Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

JAVA VIRTUAL MACHINE (JVM)

JVM stands for Java Virtual Machine that provides runtime environment to execute java byte code. JVM are most often implemented to run on existing operating system. JVM control execution of every Java program. It enables features such as automated exception handling, Garbage-collected heap.

JVM is distributed with Java class library, a set of standard class library in byte code. That implements the Java Application Programming Interface (API).

For creating desktop applications, concept of SWINGS is used.

JAVA SWINGS

Java Swing is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Some of them are explained below:

1. JButton Class

The JButton class is used to create a button that have platform-independent implementation.

2. JRadioButton Class

The JRadioButton class is used to create a radio button. It is used to choose one option from multiple options. It is widely used in exam systems or quiz. It should be added in ButtonGroup to select one radio button only.

3. JTextArea Class

The JTextArea class is used to create a text area. It is a multiline area that displays the plain text only.

4. JComboBox Class:

The JComboBox class is used to create the combobox (drop-down list). At a time only one item can be selected from the item list.

5. JColorChooser Class

The JColorChooser class is used to create a color chooser dialog box so that user can select any color.

DIFFERENCE BETWEEN AWT AND SWINGS

There are many differences between java awt and swing that are given below.

| No. | Java AWT | Java Swing |
|-----|--|---|
| 1) | AWT components are platform-dependent. | Java swing components are platform-independent. |

| | | |
|----|--|---|
| 2) | AWT components are heavyweight. | Swing components are lightweight. |
| 3) | AWT doesn't support pluggable look and feel. | Swing supports pluggable look and feel. |
| 4) | AWT provides less components than Swing. | Swing provides more powerful components such as tables, lists, scrollpanes, colorchooser, tabbedpane etc. |
| 5) | AWT doesn't follows MVC(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view. | Swing follows MVC. |

TABLE 1: DIFFERENCE BETWEEN AWT AND SWINGS

WHAT IS JFC

The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

HIERARCHY OF JAVA SWING CLASSES

The hierarchy of java swing API is given below.

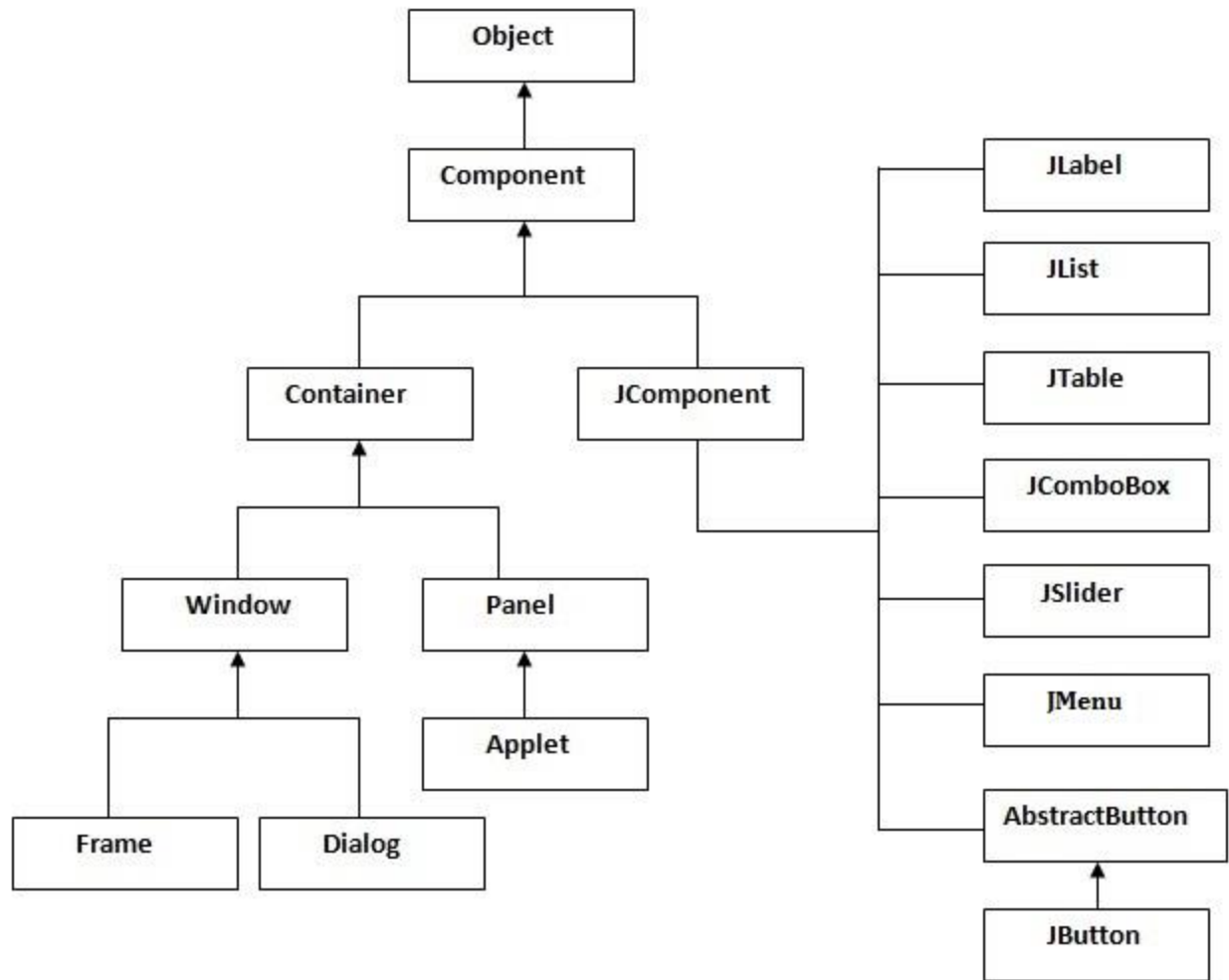


FIGURE 2: HIERARCHY OF JAVA SWING CLASSES

COMMONLY USED METHODS OF COMPONENT CLASS

The methods of Component class are widely used in java swing that are given below.

| Method | Description |
|--|---------------------------------------|
| <code>public void add(Component c)</code> | add a component on another component. |
| <code>public void setSize(int width,int height)</code> | sets size of the component. |

| | | |
|--------------------------------------|------|---|
| public setLayout(LayoutManager m) | void | sets the layout manager for the component. |
| public void setVisible(boolean b) | | sets the visibility of the component. It is by default false. |

TABLE 2: METHODS OF COMPONENT CLASS

JAVA DATABASE CONNECTIVITY (JDBC)

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

TOOL AND SOFTWARE USED FOR CREATING DESKTOP APPLICATION

DB2

DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently. DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.

History

Initially, IBM had developed DB2 product for their specific platform. Since year 1990, it decided to develop a Universal Database (UDB) DB2 Server, which can run on any authoritative operating systems such as Linux, UNIX, and Windows.

Versions

For IBM DB2, the UDB current version is 10.5 and its code name as 'Kepler'.

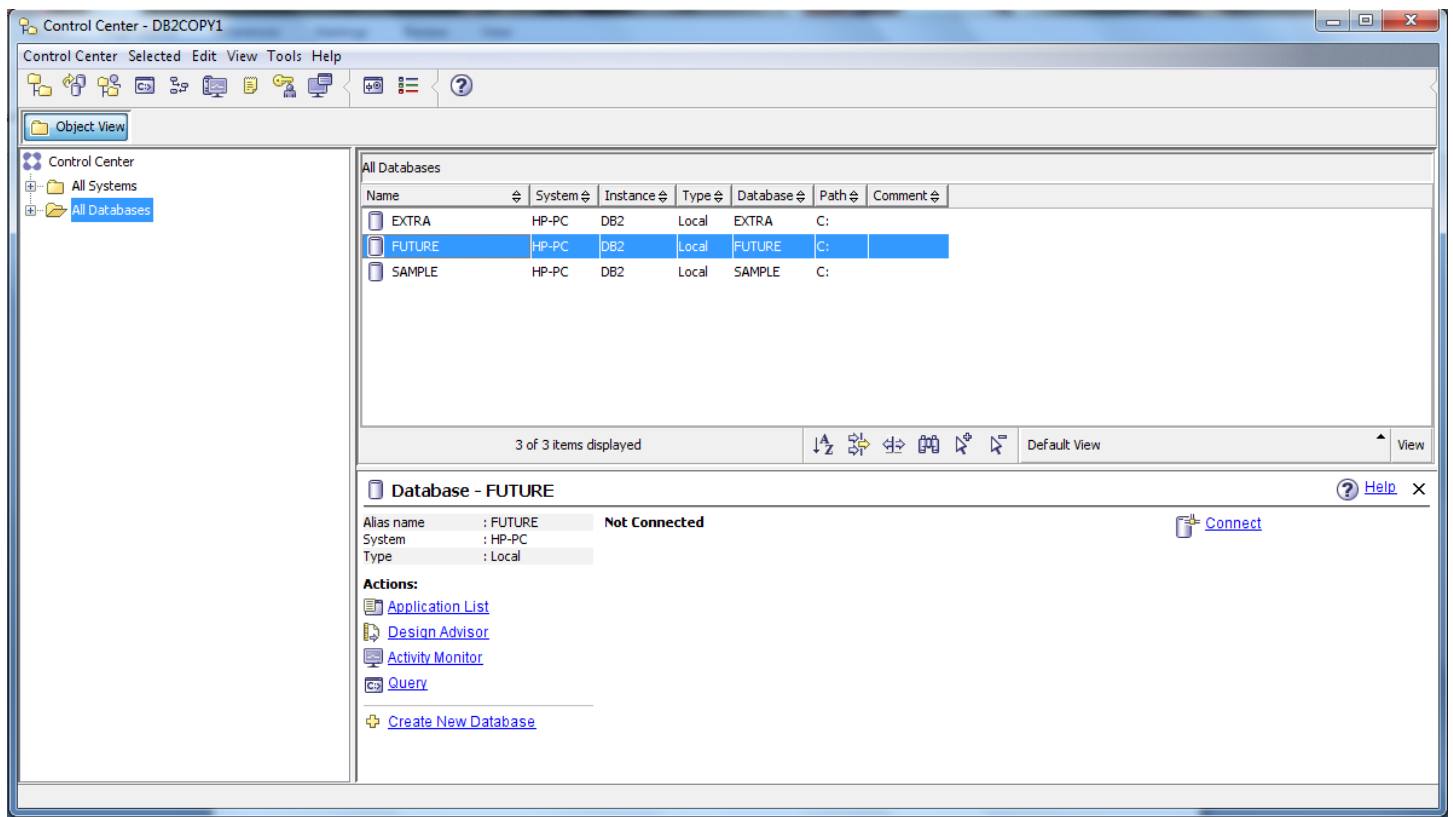
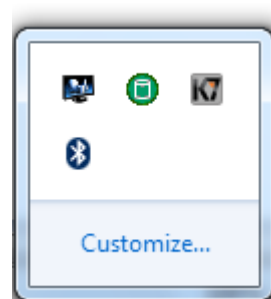


FIGURE 4: DB2 TOOL WINDOW

HOW TO START DB2:-

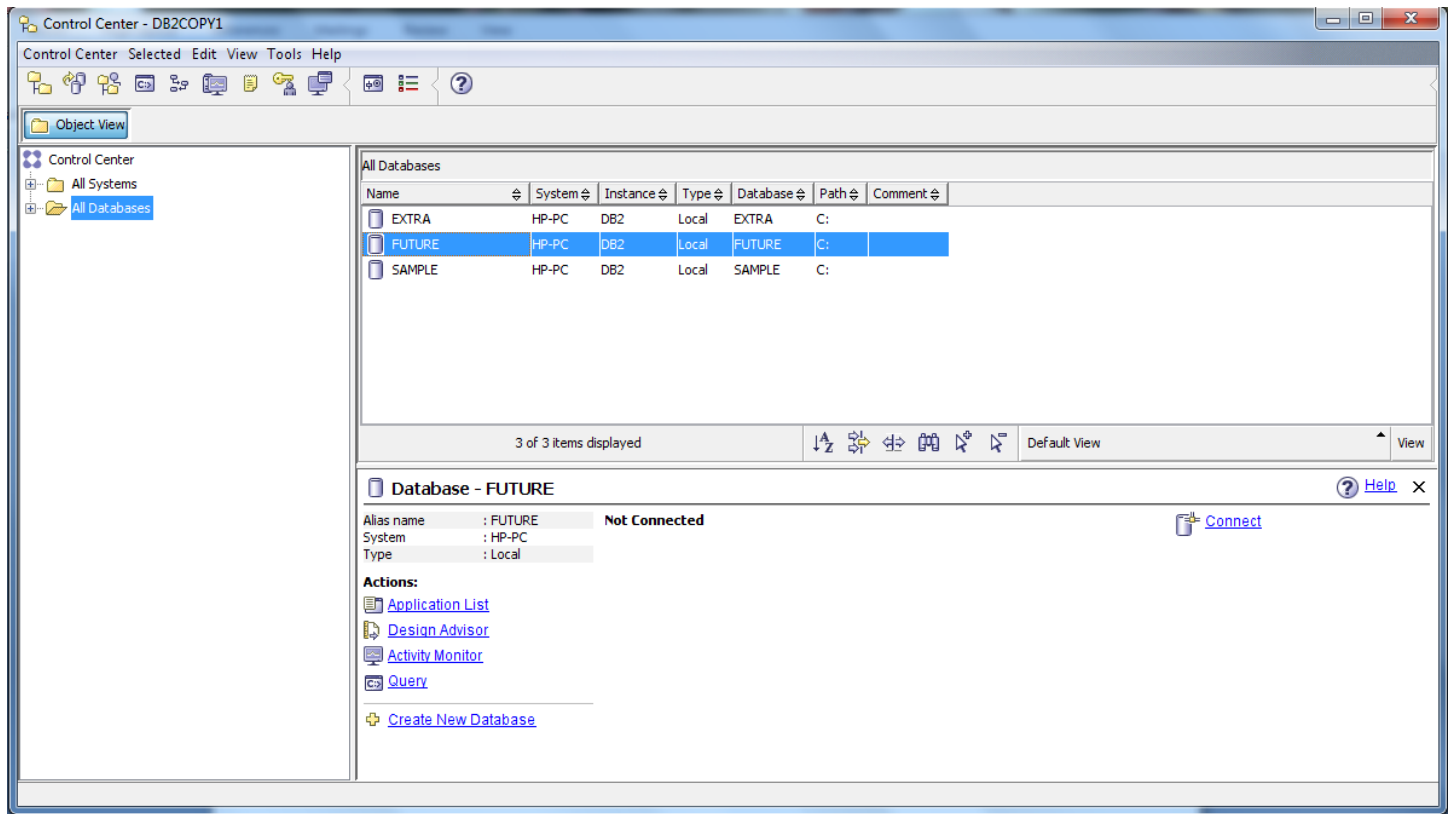
step1:-open the db2window that is created in installation.



Step2:-Goto the the green icon in picture.

Step3:-After that select the DB2control Center.

Step4:- then u see the Dialog box Given below.



step5:- The dialog box you see i.e DB2.

NETBEANS

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers. [\[3\]](#)

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.

Net Beans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The Net beans Team actively support the product and seek feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

HISTORY OF NETBEANS

Net beans begun in 1996 as xelfi ,a java IDE student project under the guidance of the faculty of Mathematics and physics at Charles university in prague. in 1997, Roman stanek formed a company around the project and produced commercial versions of the Net beans IDE until it was bought by sun Microsystems in 1999. sun open -sourced the Net beans IDE in june of the following year. since then, the net beans community has continued to grow . In 2010, sun was acquired by oracle corporation.,.

Under oracle, Net beans competed with Developer, a freeware IDE that has historically been a product of the company. In September 2016,Oracle submitted a proposal to donate the Net beans project to the Apache Software Foundation, stating that it was 'opening up the Net Beans governance model to give Net Beans constituents a greater voice in the project's direction and future success through the upcoming release of java 9 and Net beans 9 and beyond".

NETBEANS PLATFORM

The Net Beans is a framework for simplifying the development of java Swing desktop applications. The NetBeans IDE bundle for java SE contains what is needed to start developing NetBeans plugging and NetBeans platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application.

Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications,

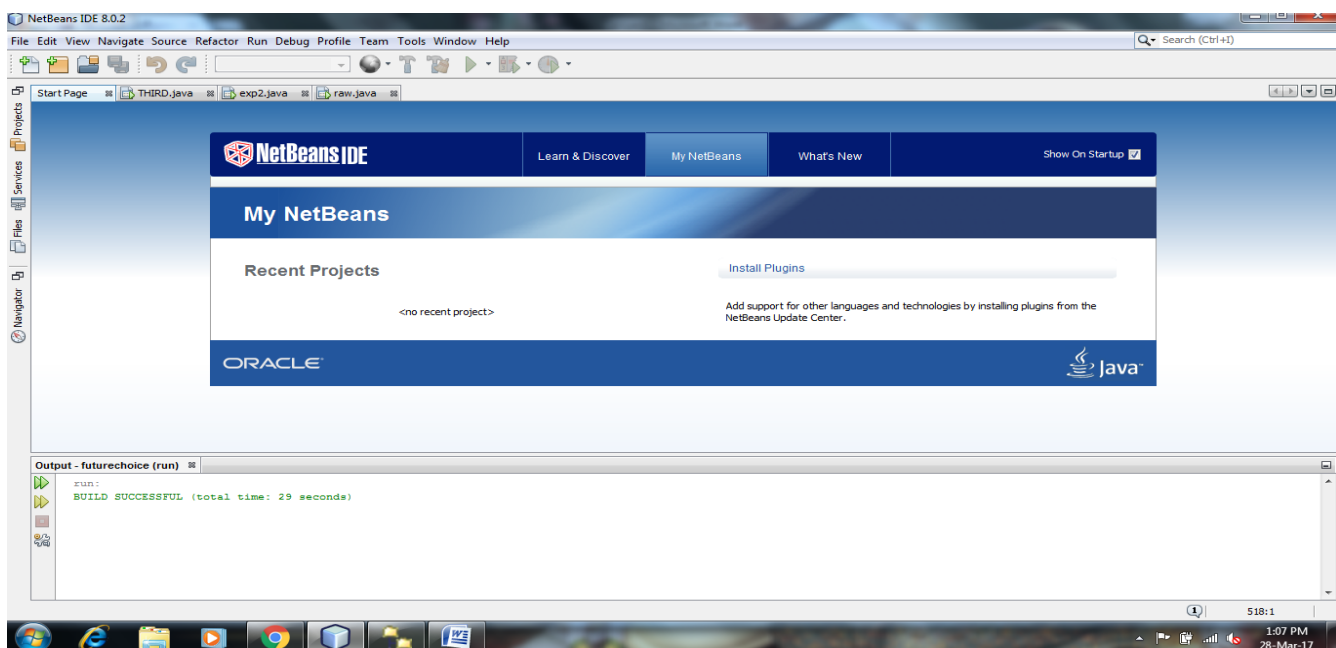


FIGURE 5: NETBEANS WINDOW

FEASIBILITY STUDY

Depending upon the result of initial investigation a more detailed feasibility study is carried out.

In this study, the requirements of the proposed system are compared with the constraints like, cost, time, manpower, technical and physical resources etc to know that all the requirements can be fulfilled in lieu of constraints or not.

Feasibility study should be conducted competently and no fundamental error of judgments are made.

Various types of feasibility study are:-

Technical Feasibility:-

In this it is checked that

- Does the necessary technology exist?
- Can the development of the proposed system be done with current equipment?
- Can the system be expanded, if developed?

Economic Feasibility:-

Economic feasibility to determine expected benefits and savings from a proposed system and compare them with costs. If the system does not appear to be profitable then it is not feasible to move ahead.

Social Feasibility:-

Social feasibility is to determine whether a proposed project will be acceptable to the people or not.

Time Feasibility:-

Time feasibility is to determine whether a proposed project can be implemented fully within a fixed time frame.

DESIGNING

This is a blueprint or a plan for a solutions for the system. here we move in the solutions domain from the problem domain. in the design phase a system is consider a set of components with clearly define a behavior and interrupt with each other in a fixed define manner to produce same behavior and service for this environments.

Type of design

1)System Design

2)Detail Design

OBJECTIVE

The objective of the design is to produced the correct design .A correct design which fulfill all the requirement decided in requirements phase.

specification of each module.

transforming Inputs to desired outputs.

PRINCIPLE

The main goal of the design is to find the best possible design within the limitations imposed by requirements and physical &social environment in which the system will operate .

Our design should be efficient effective and simple.

A design should be clearly very be complete and traceable.

CODING

CODING FOR SWITCH CASE FOR OPERATIONS

```
public void operations()
{
    switch(calculations)
    {
        case 1:
            ans=number + Double.parseDouble(jTextField1.getText());
            jTextField1.setText(Double.toString(ans));
            break;
        case 2:
            ans=number - Double.parseDouble(jTextField1.getText());
            jTextField1.setText(Double.toString(ans));
            break;
        case 3:
            ans=number * Double.parseDouble(jTextField1.getText());
            jTextField1.setText(Double.toString(ans));
            break;
        case 4:
            ans=number / Double.parseDouble(jTextField1.getText());
            jTextField1.setText(Double.toString(ans));
            break;
    }
}
```

CODING FOR PERFORMING OPERATIONS

1. ADDITION

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    number = Double.parseDouble(jTextField1.getText());  
    calculations = 1;  
    jTextField1.setText("");  
    jLabel2.setText(number + "+");  
}
```

2. SUBTRACTION

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {  
    number = Double.parseDouble(jTextField1.getText());  
    calculations = 2;  
    jTextField1.setText("");  
    jLabel2.setText(number + "-");  
}
```

3. MULTIPLICATION

```
private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {  
    number = Double.parseDouble(jTextField1.getText());  
    calculations = 3;  
    jTextField1.setText("");  
    jLabel2.setText(number + "*");  
}
```

4. DIVISION

```
private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {  
    number = Double.parseDouble(jTextField1.getText());  
    calculations = 4;  
    jTextField1.setText("");  
    jLabel2.setText(number + "/");  
}
```

5. EQUAL

```
private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {  
    operations();  
    jLabel2.setText("");  
}
```

6. DECIMAL

```
private void jButton21ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + ".");  
}
```

CODING FOR TYPING NUMBERS

1. ZERO

```
private void jButton22ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "0");  
}
```

2. ONE

```
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "1");  
}
```

3. TWO

```
private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "2");  
}
```

4. THREE

```
private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "3");  
}
```

5. FOUR

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "4");  
}
```

6. FIVE

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "5");  
}
```

7. SIX

```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "6");  
}
```



```
}
```

8. SEVEN

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "7");  
}
```

9. EIGHT

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "8");  
}
```

10. NINE

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText(jTextField1.getText() + "9");  
}
```

CODING FOR CLEAR (C) BUTTON

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText("");  
}
```

CODING FOR BACKSPACE (<) BUTTON

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    int length = jTextField1.getText().length();  
    int num = jTextField1.getText().length() - 1;  
    String store;  
    if(length>0)  
    {  
        StringBuilder backspace = new StringBuilder(jTextField1.getText());  
        backspace.deleteCharAt(num);  
        store = backspace.toString();  
        jTextField1.setText(store);  
    }  
}
```

METHODS FOR ON (ENABLE) AND OFF (DISABLE) BUTTON

```
public void disable(){  
    jTextField1.setEnabled(false);  
    jRadioButton1.setEnabled(true);  
    jRadioButton2.setEnabled(false);  
    jButton1.setEnabled(false);  
    jButton2.setEnabled(false);  
    jButton3.setEnabled(false);  
    jButton4.setEnabled(false);  
    jButton5.setEnabled(false);  
    jButton6.setEnabled(false);  
    jButton7.setEnabled(false);  
    jButton8.setEnabled(false);  
    jButton9.setEnabled(false);  
    jButton10.setEnabled(false);  
    jButton11.setEnabled(false);  
    jButton12.setEnabled(false);  
    jButton13.setEnabled(false);  
    jButton14.setEnabled(false);  
    jButton15.setEnabled(false);  
    jButton16.setEnabled(false);  
    jButton17.setEnabled(false);  
    jButton18.setEnabled(false);  
    jButton19.setEnabled(false);  
    jButton20.setEnabled(false);  
}
```

```
jButton21.setEnabled(false);  
jButton22.setEnabled(false);  
}
```

```
public void enable(){  
    jTextField1.setEnabled(true);  
    jRadioButton1.setEnabled(false);  
    jRadioButton2.setEnabled(true);  
    jButton1.setEnabled(true);  
    jButton2.setEnabled(true);  
    jButton3.setEnabled(true);  
    jButton4.setEnabled(true);  
    jButton5.setEnabled(true);  
    jButton6.setEnabled(true);  
    jButton7.setEnabled(true);  
    jButton8.setEnabled(true);  
    jButton9.setEnabled(true);  
    jButton10.setEnabled(true);  
    jButton11.setEnabled(true);  
    jButton12.setEnabled(true);  
    jButton13.setEnabled(true);  
    jButton14.setEnabled(true);  
    jButton15.setEnabled(true);  
    jButton16.setEnabled(true);  
    jButton17.setEnabled(true);  
}
```

```
jButton18.setEnabled(true);  
jButton19.setEnabled(true);  
jButton20.setEnabled(true);  
jButton21.setEnabled(true);  
jButton22.setEnabled(true);  
}
```

CODING FOR ON AND OFF BUTTON

1. ON

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    enable();  
}
```

2. OFF

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    disable();  
}
```

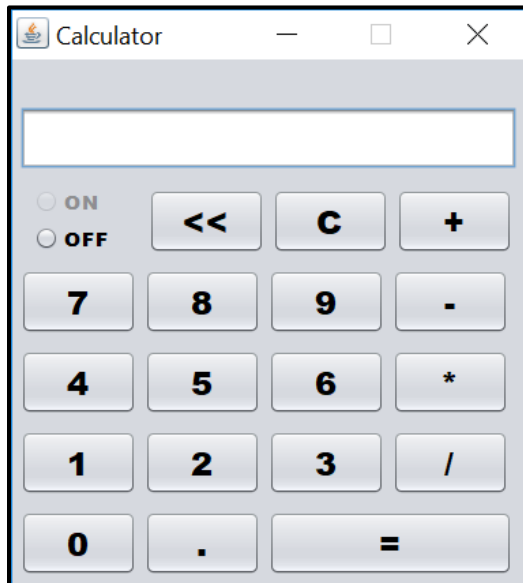
FEATURES

The Calculator has the following features:

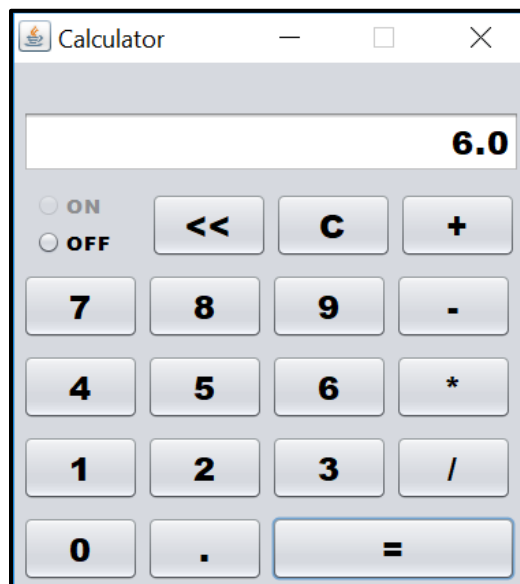
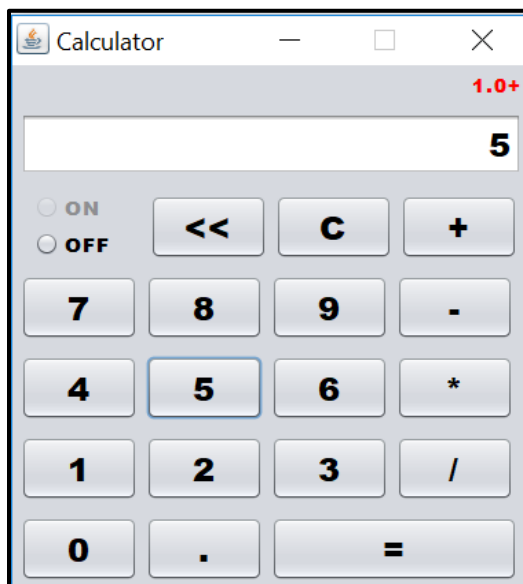
1. **Number Buttons** to enter Numbers which they show on the Text Field
2. **Operational Buttons** to perform operations like addition, subtraction, multiplication and division
3. **Equal Button** shows the result after performing operation on numbers
4. **Clear Button** to clear the Text Field for new calculation
5. **Backspace Button** to delete entered values
6. **Off Button** to disable all the buttons and text field
7. **On Button** to enable all the buttons and text field
8. **Calculator Frame** is non resizable

WORKING

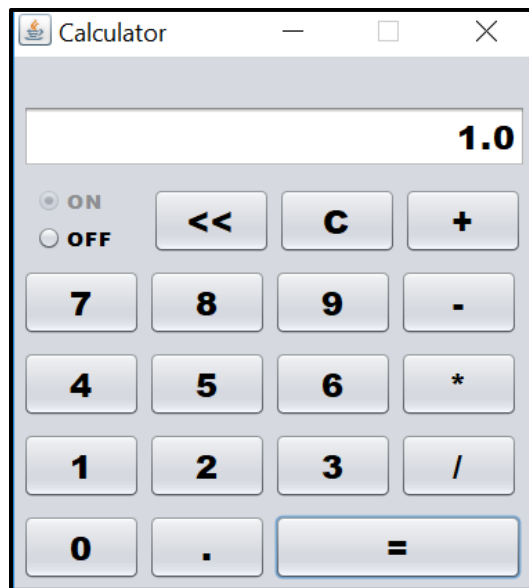
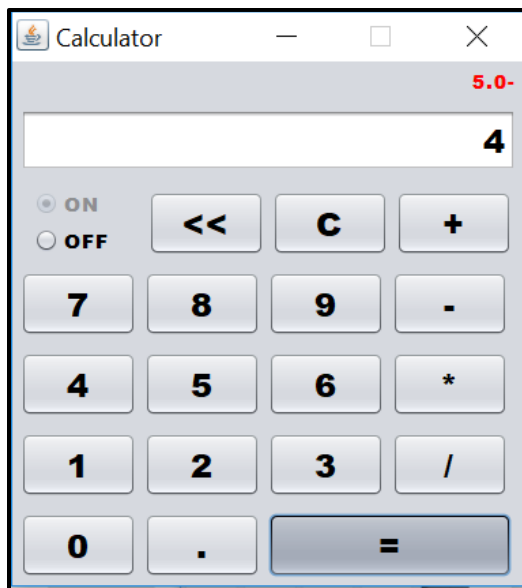
1. DEFAULT SCREEN



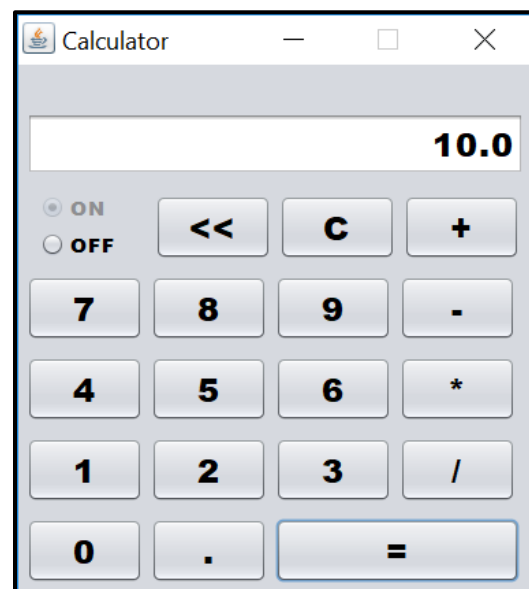
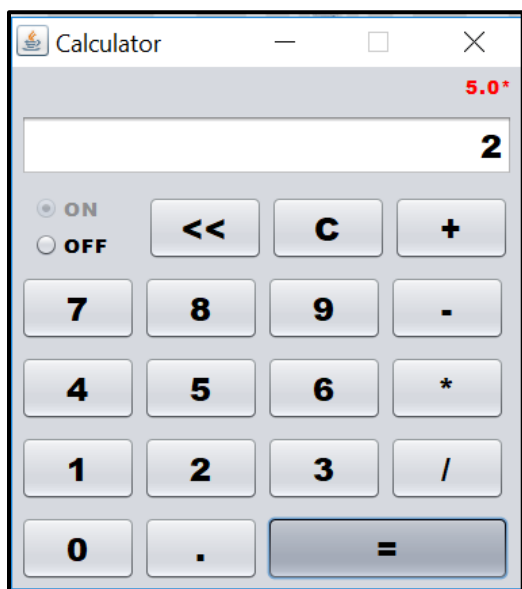
2. ADDITION OPERATION



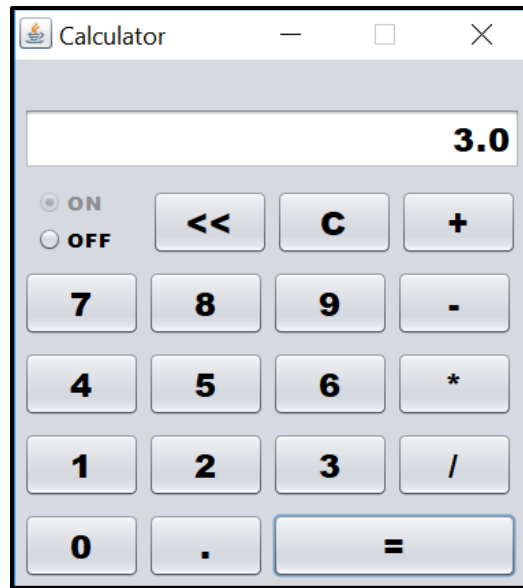
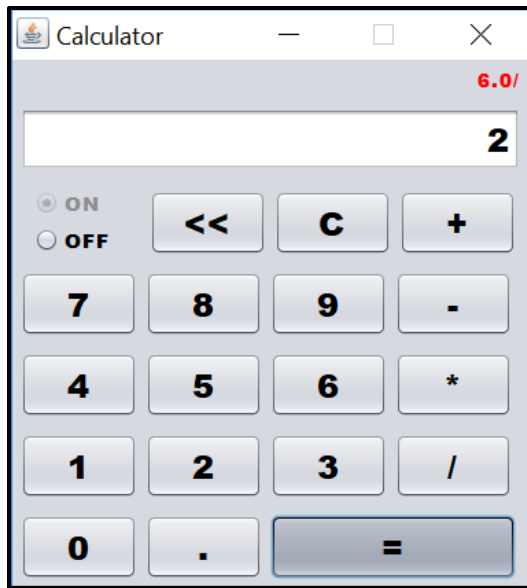
3. SUBTRACTION OPERATION



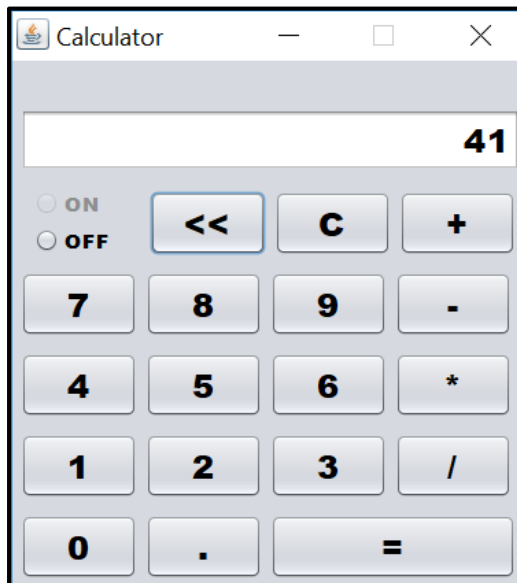
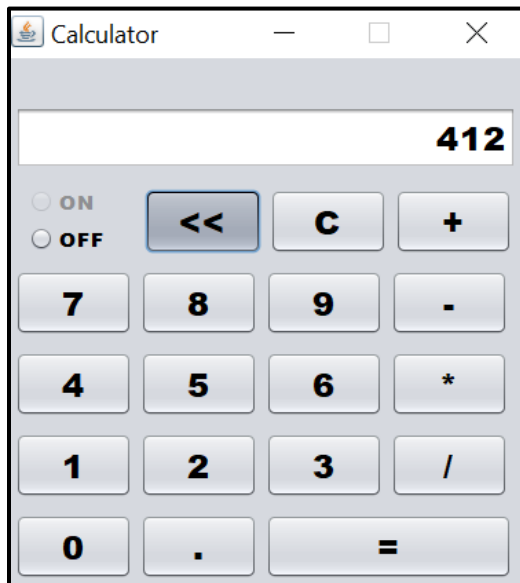
4. MULTIPLICATION OPERATION



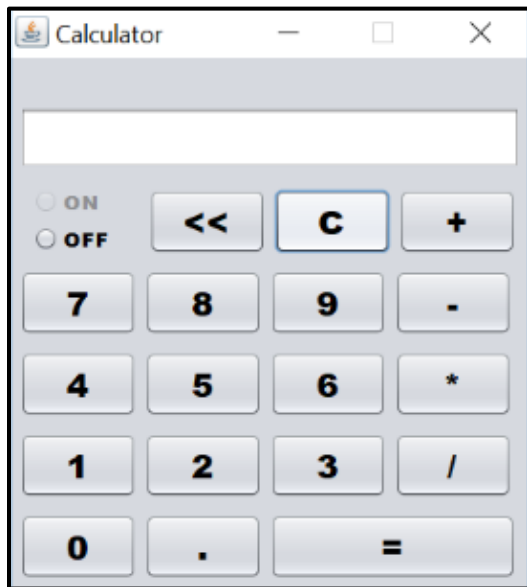
5. DIVISION OPERATION



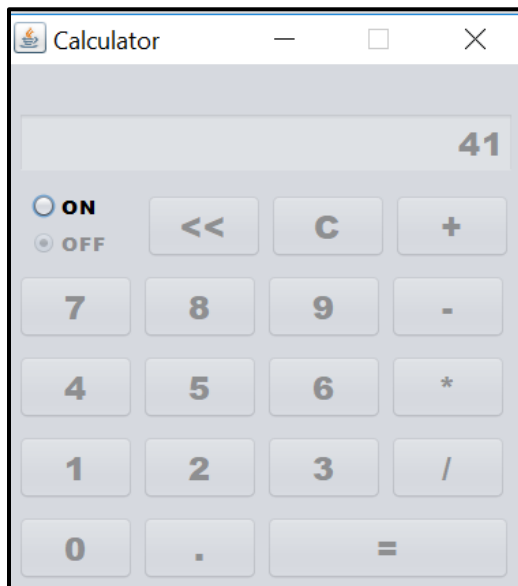
6. BACKSPACE BUTTON



7. CLEAR BUTTON



8. OFF BUTTON



9. ON BUTTON

