

Diploma Engineering

Laboratory Manual

(Mobile Application Development)

(DI04016051)

[Diploma Information Technology Semester-4]

Enrollment No	
Name	
Branch	
Academic Term	
Institute	



Directorate Of Technical Education
Gandhinagar - Gujarat

DTE's Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

DTE's Mission:

Institute's Vision:

Student should write

Institute's Mission:

Student should write

Department's Vision:

Student should write

Department's Mission:

Student should write

Certificate

This is to certify that Mr./Ms
Enrollment No. of Semester of *Diploma*
in of (GTU Code) has
satisfactorily completed the term work in course
..... for the academic
year: Term: Odd/Even prescribed in the GTU curriculum.

Place:.....

Date:

Signature of Course Faculty

Head of the Department

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum -2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities for each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

Mobile applications have become an essential component of businesses across industries in today's technology-driven world. There is a growing demand for skilled Mobile application developers capable of developing innovative and robust applications that meet the needs of both businesses and end users. This course develops necessary skills in students, after learning this course; students will be able to develop user-friendly mobile applications.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis:** Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

Practical Outcome - Course Outcome matrix

Course Outcomes (COs): <ul style="list-style-type: none"> a. <u><i>CO1: Understand the fundamentals of Android OS.</i></u> b. <u><i>CO2: Understand Android architecture and use its core components.</i></u> c. <u><i>CO3: Develop interactive User Interfaces using layouts and views.</i></u> d. <u><i>CO4: Implement data storage using SQLite and Firebase Realtime Database.</i></u> e. <u><i>CO5: Explore and demonstrate modern technologies such as Kotlin, Flutter (Cross-platform development), APIs, and JSON parsing.</i></u> 						
S. No.	Practical Outcome/Title of experiment	CO1	CO2	CO3	CO4	CO5
1	Install Android Studio, configure SDK, and create a simple “Hello World” application.	√	-	-	-	-
2	Develop a simple app that demonstrates the activity lifecycle.	-	√	-	-	-
3	Design a static user interface using Linear Layout and Relative Layout.	-	-	√	-	-
4	Design a user interface using Constraint Layout and Grid Layout.	-	-	√	-	-
5	Develop a simple calculator app that takes user input and performs basic arithmetic operations like addition, subtraction, multiplication, and division.	-	-	√	-	-
6	Develop an Android application that uses an Intent to pass data between different activities.	-	-	√	-	-
7	Develop an Android application that uses Implicit intent to open the dialer with a given phone number.	-	-	√	-	-
8	Develop an Android application to change screen color as per the user choice from a menu.	-	-	√	-	-
9	Develop an Android application that uses Content Providers to share data between different apps and components.	-	√	-	-	-
10	Develop an Android application that stores and retrieves username using shared preferences.	-	-	-	√	-

11	Develop an Android application that creates a database using SQLiteOpenHelper Class and performs Insert and Read from the SQLite database.	-	-	-	✓	-
12	Develop an Android application to Update and Delete data from the SQLite database using SQ Lite Open Helper class.	-	-	-	✓	-
13	Develop an Android application that integrates Firebase real-time database and store the data into it.	-	-	-	✓	-
14	Develop an Android application to retrieve data from Firebase Realtime Database or a JSON data source (REST API), parse the JSON response, and display the data in a Recycler View.	-	-	-	-	✓

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

Develop user-friendly Mobile applications, design intuitive user interfaces, understand and implement various app components, effectively debug and troubleshoot issues, adapt to emerging technologies and continuously enhance their skills to meet the demands of the industry.

Guidelines to Course Faculty

1. Course faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Course faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should Involve & give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Students shall develop maintenance skill as expected by industries.
2. Student shall attempt to develop related hand-on skills and build confidence.
3. Student shall develop the habits of evolving more ideas, innovations, skills etc.
4. Student shall refer technical magazines and data books.
5. Student must take print out of the code and attach it with every practical.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Continuous Assessment Sheet

Enrolment No:**Student Name:**

Sr no	Practical Outcome/Title of Experiment	Page	Date	Marks (25)	Sign
1	Install Android Studio, configure SDK, and create a simple “Hello World” application.				
2	Develop a simple app that demonstrates the activity lifecycle.				
3	Design a static user interface using Linear Layout and Relative Layout.				
4	Design a user interface using Constraint Layout and Grid Layout.				
5	Develop a simple calculator app that takes user input and performs basic arithmetic operations like addition, subtraction, multiplication, and division.				
6	Develop an Android application that uses an Intent to pass data between different activities.				
7	Develop an Android application that uses Implicit intent to open the dialer with a given phone number.				
8	Develop an Android application to change screen color as per the user choice from a menu.				
9	Develop an Android application that uses Content Providers to share data between different apps and components.				
10	Develop an Android application that stores and retrieves username using shared preferences.				
11	Develop an Android application that creates a database using SQLiteOpenHelper Class and performs Insert and Read from the SQLite database.				
12	Develop an Android application to Update and Delete data from the SQLite database using SQ Lite Open Helper class.				
13	Develop an Android application that integrates Firebase real-time database and store the data into it.				
14	Develop an Android application to retrieve data from Firebase Realtime Database or a JSON data source (REST API), parse the JSON response, and display the data in a Recycler View.				
Total					

Date:

Practical No.1: Install Android Studio, configure SDK, and create a simple “Hello World” application.

A. Objective:

To install and configure the Android Studio and development environment.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Be able to apply engineering knowledge of computing appropriate to the problem.
2. **Engineering Tools, Experimentation and Testing:** Be able to use and apply current technical concepts and best practices in information technologies.
3. **Engineering practices for society, sustainability, and environment:** Apply appropriate technology in context of society, sustainability, environment, and ethical practices.
4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: ‘Configure Android Studio for mobile application development’

D. Expected Course Outcomes(COs)

CO1: Understand the fundamentals of Android OS.

E. Practical Outcome(PRo)

Set up the Android development environment, and create a simple "Hello World" app.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

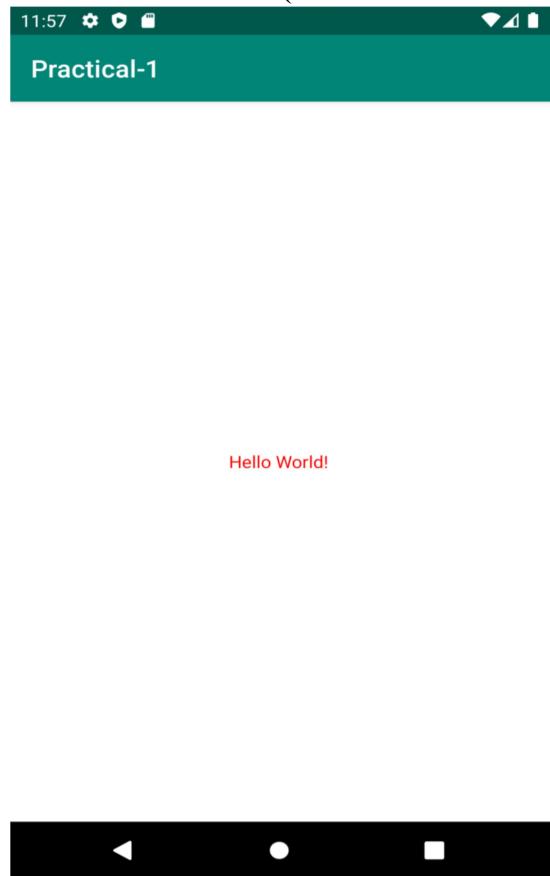
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Download latest version of Android Studio.
2. Install Android Studio.
3. Start Android Studio IDE.
4. Make a new project with empty activity, click next.
5. Give a Project name, Select the language for app development (Java) and choose a minimum SDK for the application and then click on Finish.
6. Go to the AVD manager and create a virtual device or connect your mobile device via USB for running/debugging your app.
7. Go to activity_main.xml file. Take a TextView from the palette and change the text attribute to “Hello World”.
8. Click on the Run button in the tool bar and check the output.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an AVD? How do you create an AVD in Android Studio?

2. List the different versions of Android.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.2: Develop a simple app that demonstrates the activity lifecycle.

A. Objective:

To understand the activity lifecycle of an Android app.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Be able to apply engineering knowledge of computing appropriate to the problem.
2. **Engineering Tools, Experimentation and Testing:** Be able to use and apply current technical concepts and best practices in information technologies.
3. **Engineering practices for society, sustainability, and environment:** Apply appropriate technology in context of society, sustainability, environment, and ethical practices.
4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Understand activity life cycle for mobile application development'

D. Expected Course Outcomes(Cos)

CO2: Understand Android architecture and use its core components.

E. Practical Outcome(PRo)

Understand the activity lifecycle and create a simple app to demonstrate the various methods in activity lifecycle.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

The Android Activity lifecycle refers to the sequence of states that an activity goes through during its lifetime, from creation to destruction. It is crucial to understand the activity lifecycle when developing Android applications, as it enables you to manage and control the behavior of your app's user interface and data.

An activity is a fundamental component of an Android application, representing a single screen with a user interface. Activities have various states, and as the user interacts with the app, they transition between these states. The Android platform provides a set of lifecycle callback methods that you can override to perform specific actions at each stage of the activity's lifecycle.

The key states in the activity lifecycle include:

onCreate(): This is the first method called when an activity is created. It is where you initialize essential components, such as layout, views, and data.

onStart(): This method is called when the activity becomes visible to the user but is not yet in the foreground. It prepares the activity to interact with the user.

onResume(): When the activity enters the foreground and becomes interactive, this method is called. It is an excellent place to start animations, play audio/video, and initialize sensors.

onPause(): This method is invoked when the activity loses focus and becomes partially visible but still partially interactive. It is a good place to save data and release resources that are not needed while the activity is not in the foreground.

onStop(): When the activity is no longer visible to the user, this method is called. It provides an opportunity to clean up resources that should be released when the activity is not in use.

onDestroy(): This is the final method called before the activity is destroyed. It allows you to perform any cleanup operations, such as releasing resources, unregistering listeners, etc.

Understanding the activity lifecycle helps you handle various scenarios, such as saving and restoring instance state during configuration changes (e.g., screen rotation), gracefully handling interruptions like phone calls or other app notifications, and managing system resources efficiently.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

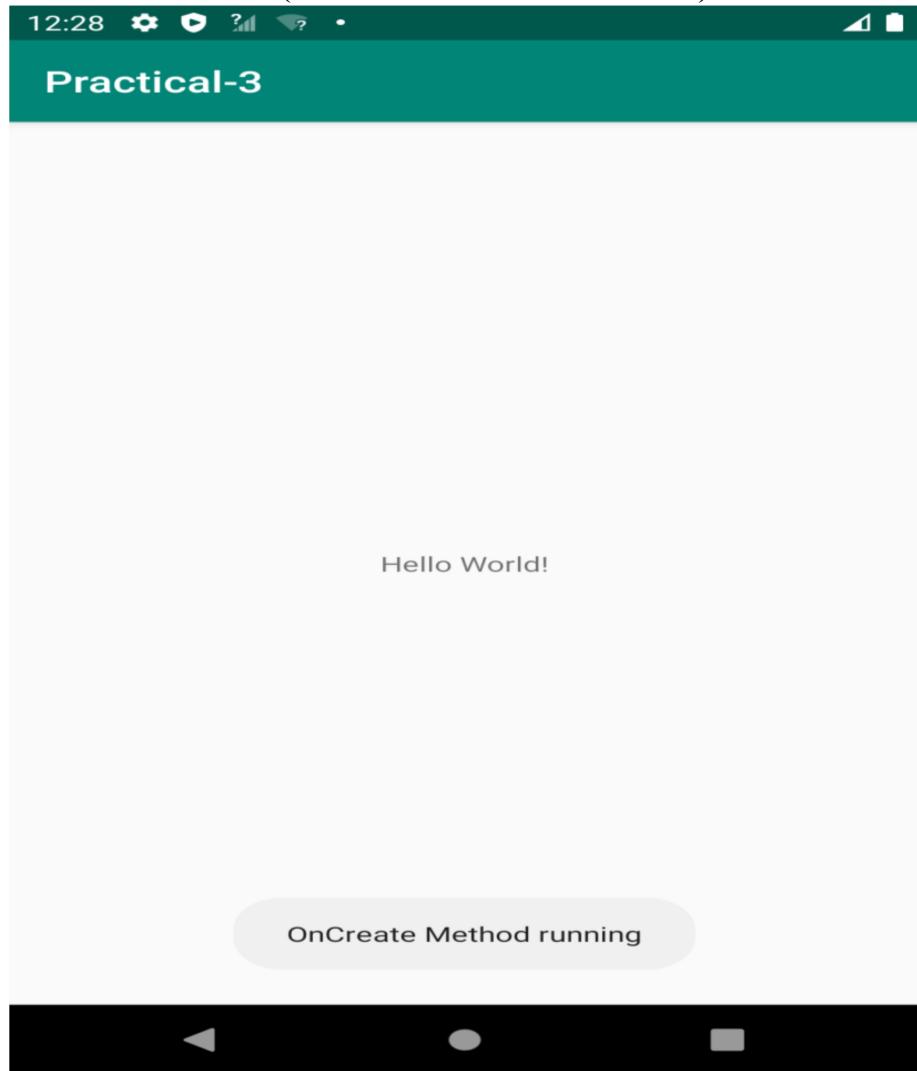
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Open the main activity file (MainActivity.java) and locate the generated lifecycle callback methods.
6. Implement code within each lifecycle method to log a message or display a Toast notification indicating the current lifecycle state.
7. Run the application on an Android emulator or a physical device.
8. Observe the logcat output or Toast notifications as you interact with the application and switch between different states.
9. Test scenarios like rotating the screen, switching between apps, or receiving phone calls to observe the activity lifecycle behavior.
10. Analyze the results and observe how the activity transitions through its lifecycle states.
11. Modify the app or add additional features to further explore the activity lifecycle if desired.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



Note: Take screenshots of every method in the Activity Lifecycle.

M. Practical related Quiz.

1. Explain the concept of the Android activity lifecycle and its significance in Android app development?

2. Describe the sequence of lifecycle callback methods that are invoked when an activity is created and initialized.
 3. What are the key states in the activity lifecycle, and what are the purposes of each state?

N. References / Suggestions

1. <https://developer.android.com>
 2. <https://www.javatpoint.com/android-tutorial>
 3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.3: Design a static user interface using Linear Layout and Relative Layout.

A. Objective:

To understand the usage of Linear Layout and Relative Layout in an Android app.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Be able to apply engineering knowledge of computing appropriate to the problem.
2. **Engineering Tools, Experimentation and Testing:** Be able to use and apply current technical concepts and best practices in information technologies.
3. **Engineering practices for society, sustainability, and environment:** Apply appropriate technology in context of society, sustainability, environment, and ethical practices.
4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Design a user interface using Linear Layout and Relative Layout'

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand the layouts and create a simple application in android.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

(A) Linear Layout

In Android, **LinearLayout** is a ViewGroup that arranges its child views either vertically or horizontally in a linear fashion. It is one of the most used layout managers for designing user interfaces.

Key Concepts:

Orientation: LinearLayout supports two orientations: vertical and horizontal. The orientation determines how the child views are arranged. In the vertical orientation,

the child views are stacked vertically from top to bottom, while in the horizontal orientation, they are placed horizontally from left to right.

Layout_weight: LinearLayout allows you to assign a weight to child views using the layout_weight attribute. The weight determines how the available space is distributed among the child views. By assigning weights, you can control how much space each view occupies relative to others.

Gravity and Layout_gravity: The gravity attribute defines how the child views are aligned within the LinearLayout. It can be set to values such as top, bottom, start, end, center, etc. On the other hand, the layout_gravity attribute determines the alignment of the LinearLayout itself within its parent view.

(B) Relative Layout

In Android, **RelativeLayout** is a ViewGroup that allows you to arrange UI components relative to each other. It offers a flexible and powerful way to position and align views based on their relationships.

Key Concepts:

Relative positioning: With RelativeLayout, you can define the position of a view relative to another view using attributes such as "layout_above," "layout_below," "layout_toStartOf," "layout_toEndOf," etc. These attributes specify the relationships between views and determine their placement.

Anchoring: RelativeLayout allows you to anchor a view to a specific position within its parent or to another view. Attributes like "layout_alignParentTop," "layout_alignParentBottom," "layout_alignStart," "layout_alignEnd," etc., control the anchoring behavior.

Centering: RelativeLayout provides attributes such as "layout_centerHorizontal" and "layout_centerVertical" to center a view horizontally or vertically within its parent.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

(A) Linear Layout

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Open the layout file (activity_main.xml) and design the calculator's user interface using appropriate widgets, such as TextViews for display and Buttons for number and operation input.
6. Define appropriate IDs for each widget to reference them in the Java code.
7. Open the main activity file (MainActivity.java) and create references to the relevant widgets using findViewById() method.
8. Implement event listeners for the number and operation buttons to capture user input.
9. Write the logic to perform the arithmetic operations based on the user's input, including addition, subtraction, multiplication, and division.
10. Handle error cases, such as dividing by zero or invalid input, and provide appropriate feedback to the user.
11. Update the result TextView with the calculated value.
12. Run the application on an Android emulator or a physical device.
13. Test the calculator app by entering various numbers and performing different arithmetic operations.
14. Verify that the app displays the correct results and handles errors gracefully.
15. Optionally, enhance the app by adding additional features, such as a clear button or support for decimals.

(B) Relative Layout

1. Open Android Studio and create a new Android project.

2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Open the layout file (activity_main.xml) and design the user interface using a RelativeLayout as the root element.
6. Add the desired UI components (such as buttons, text views, or image views) within the RelativeLayout, specifying their attributes such as width, height, and margins.
7. Use RelativeLayout attributes (e.g., "layout_above," "layout_below," "layout_toStartOf," etc.) to position and align the views relative to each other.
8. Utilize anchoring attributes (e.g., "layout_alignParentTop," "layout_alignParentBottom," "layout_alignStart," etc.) to anchor views to specific positions within the parent or to other views.
9. If needed, apply centering attributes (e.g., "layout_centerHorizontal," "layout_centerVertical") to center views horizontally or vertically within the RelativeLayout.
10. Run the application on an Android emulator or a physical device.
11. Test the app and verify that the UI components are arranged and aligned based on the specified RelativeLayout attributes.
12. Make adjustments to the RelativeLayout attributes and UI components as needed to achieve the desired UI layout.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):

Linear Layout



Relative Layout



M. Practical related Quiz.

1. Describe the layout_gravity attribute in LinearLayout and how it determines the alignment of the LinearLayout itself within its parent view?

2. How would you align the child views within a LinearLayout at the center horizontally and vertically?

3. How do RelativeLayout attributes like "layout_above" and "layout_below" help in positioning views relative to each other?
 4. Describe how you would create a UI layout in RelativeLayout where one view is centered horizontally and vertically within its parent.

N. References / Suggestions

1. <https://developer.android.com>
 2. <https://www.javatpoint.com/android-tutorial>
 3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.4: Design a user interface using Constraint Layout and Grid Layout.

A. Objective:

To understand the usage of Constraint Layout and Grid Layout in an Android application.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Design responsive and structured user interfaces using Constraint Layout and Grid Layout'

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand the layouts and create a simple Android application using Constraint Layout and Grid Layout.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

(A) Constraint Layout

ConstraintLayout is an advanced Android layout that allows positioning and sizing of UI components using constraints instead of nesting multiple layouts. Each view is positioned relative to the parent layout or other views.

Key Concepts:

Constraints: Define how a view is aligned (top, bottom, start, end) relative to another view or parent.

Bias: Controls positioning when constraints are applied on both sides.

Flat View Hierarchy: Improves performance compared to nested layouts.

Responsive Design: Automatically adjusts UI for different screen sizes.

ConstraintLayout is widely used in modern Android applications for building flexible and responsive user interfaces.

(B) Grid Layout

GridLayout arranges UI components in a grid format of rows and columns, similar to a table structure.

Key Concepts:

Row and Column Count: Defines the grid structure.

Cell Placement: Each child view occupies a specific cell.

RowSpan and ColumnSpan: Allows views to span across multiple rows or columns.

Structured UI: Useful for calculators, dashboards, and form-based layouts.

GridLayout provides an organized and predictable layout arrangement.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

(A) Constraint Layout

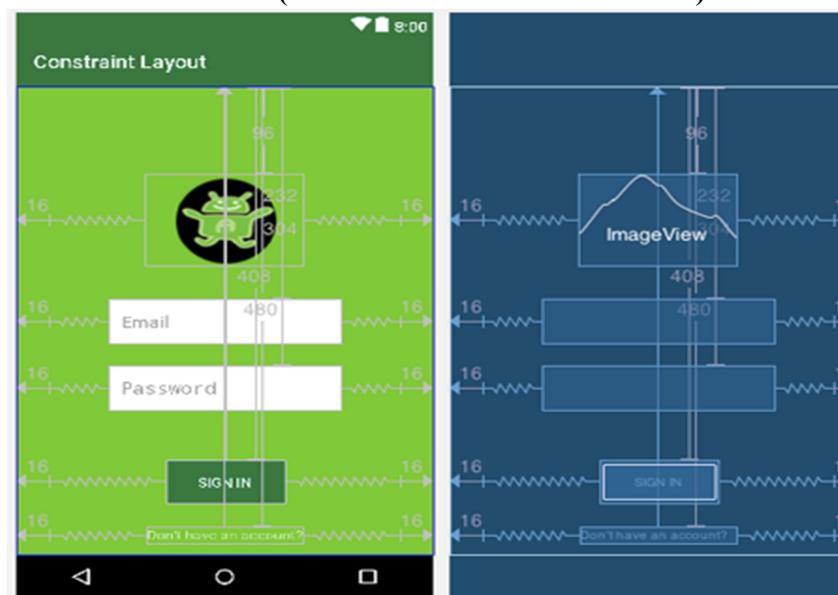
1. Open **Android Studio** and create a new Android project.
2. Provide a suitable project name and select the minimum SDK.
3. Choose **Empty Activity** and click **Finish**.
4. Open `activity_main.xml`.
5. Set the root layout as **ConstraintLayout**.
6. Add UI components such as `TextView` and `Button`.
7. Apply constraints (top, bottom, start, end) to position the views.
8. Adjust margins and bias for proper alignment.
9. Run the application on Emulator or physical device.
10. Verify that the UI adjusts properly on screen.

(B) Grid Layout

1. Create a new layout file `grid_layout.xml`.
2. Set the root element as `GridLayout`.
3. Define `rowCount` and `columnCount`.
4. Add Buttons or `TextViews` inside the grid.
5. Run the application and observe grid arrangement.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):





M. Practical related Quiz.

1. What is ConstraintLayout and why is it preferred over nested layouts?
2. What is the purpose of constraints in ConstraintLayout?
3. How does GridLayout arrange UI components?

4. Mention two applications where GridLayout is useful.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.5: Develop a simple calculator app that takes user input and performs basic arithmetic operations like addition, subtraction, multiplication, and division.

A. Objective:

To understand the basic layouts and components of an Android app.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: ‘Design a simple calculator using layouts and android components’

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand the layouts and create a simple calculator in android.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

To develop a calculator app in Android Studio, you'll need to have a solid understanding of the following concepts and components:

Basic Java programming: Familiarity with Java programming language is necessary for implementing the logic of the calculator operations and handling user input.

Android Studio: You should be familiar with the Android Studio Integrated Development Environment (IDE), including how to create a new Android project, navigate the project structure, and use the layout editor.

XML Layouts: Understanding XML layout files is essential for designing the user interface of the calculator app. You'll use various layout components, such as TextViews, EditTexts, Buttons, and LinearLayouts, to create a visually appealing and functional UI.

Event Handling: Knowledge of event handling is crucial for capturing user input and responding to button clicks. You'll need to implement event listeners to detect button clicks and perform the corresponding arithmetic operations.

Android Widgets: Understanding how to use various Android widgets, such as TextViews and EditTexts, to display and receive user input is important for creating an interactive calculator app.

Mathematical Operations: A solid grasp of basic arithmetic operations, including addition, subtraction, multiplication, and division, is necessary to implement the calculator's logic accurately.

Error Handling: It's important to handle potential errors, such as dividing by zero or invalid input, to ensure the calculator app functions correctly and provides appropriate feedback to the user.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	1
2.	<u>Android Studio</u>	<u>Latest Version</u>	1

J. Safety and necessary Precautions followed

1. Turn off the power switch only after the computer is shut down.

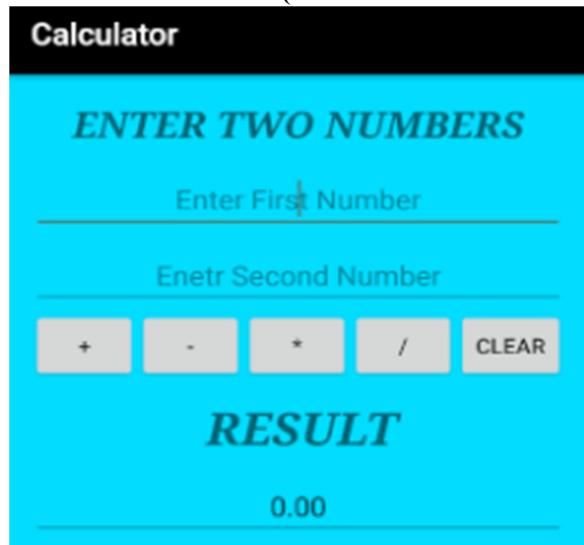
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Open the layout file (activity_main.xml) and design the calculator's user interface using appropriate widgets, such as TextViews for display and Buttons for number and operation input.
6. Define appropriate IDs for each widget to reference them in the Java code.
7. Open the main activity file (MainActivity.java) and create references to the relevant widgets using findViewById() method.
8. Implement event listeners for the number and operation buttons to capture user input.
9. Write the logic to perform the arithmetic operations based on the user's input, including addition, subtraction, multiplication, and division.
10. Handle error cases, such as dividing by zero or invalid input, and provide appropriate feedback to the user.
11. Update the result TextView with the calculated value.
12. Run the application on an Android emulator or a physical device.
13. Test the calculator app by entering various numbers and performing different arithmetic operations.
14. Verify that the app displays the correct results and handles errors gracefully.
15. Optionally, enhance the app by adding additional features, such as a clear button or support for decimals.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. How would you design the user interface of a calculator app to ensure a user-friendly and intuitive experience?
2. What event listener would you use to capture button clicks in the calculator app, and how would you implement it?
3. Describe the steps you would take to implement the addition functionality in the calculator app.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.6: Develop an Android application that uses an Intent to pass data between different activities.

A. Objective:

To understand the usage of Intents in Android.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis:** Identify and analyse well-defined engineering problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Design a simple application using Intents to pass data between different activities'

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand the Intents and design a simple application to switch and pass data between different activities in android.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Android, Intents are a fundamental mechanism for communication between different components of an application, such as activities, services, and broadcast receivers. They allow you to start new activities, launch services, and send broadcast messages.

Key Concepts:

Intent: An Intent is a messaging object that represents an "intention" to perform an action. It can be used to start activities, launch services, and broadcast messages. Intents can carry data between components through extras.

Explicit Intent: An Explicit Intent is used to launch a specific component within the same application. It explicitly specifies the target component's name or class.

Implicit Intent: An Implicit Intent is used to request an action from components in other applications. It specifies the action to be performed and optionally the data it operates on.

Extras: Extras are key-value pairs that can be added to an Intent to carry additional data. They are used to pass information from one activity to another.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

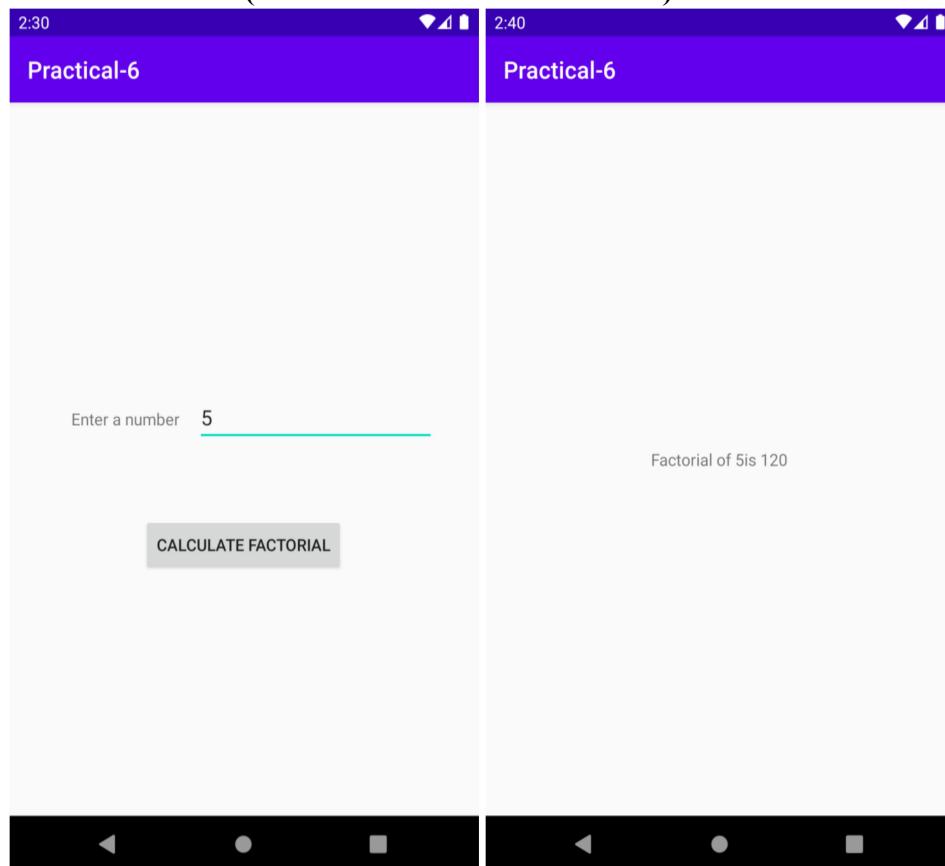
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Create two or more activities that you want to navigate between and pass data.
6. Design the user interfaces for each activity using XML layout files.
7. In the source activity, create an Intent object and add extras to it to pass the data.
8. Start the target activity using `startActivity()` or `startActivityForResult()` methods, passing the Intent as an argument.
9. In the target activity, retrieve the data passed from the source activity using `getIntent()` and `getIntent().getExtras()` methods.
10. Display or use the retrieved data in the target activity as desired.
11. Run the application on an Android emulator or a physical device.
12. Test the app by navigating between activities and verifying that the data is successfully passed and displayed.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is the purpose of using Intents in Android applications?
 2. Explain the difference between Explicit Intents and Implicit Intents.
 3. How do you pass data between activities using Intents?
 4. How do you retrieve the data passed from one activity to another using Intent?

N. References / Suggestions

1. <https://developer.android.com>
 2. <https://www.javatpoint.com/android-tutorial>
 3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.7: Develop an Android application that uses Implicit intent to open the dialer with a given phone number

A. Objective:

To understand the use of Implicit Intent in Android by opening the phone dialer with a given phone number.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Use Implicit Intent for inter-application communication in Android'

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand implicit intents and create an Android application to open the dialer with a predefined phone.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Android, an Intent is a messaging object used to request an action from another app component. Intents are mainly of two types: Explicit Intent and Implicit Intent.

An Implicit Intent does not specify the target component directly. Instead, it declares a general action to perform, allowing the Android system to find a suitable application to handle the request.

To open the phone dialer, Android provides the predefined action Intent.ACTION_DIAL, which opens the dialer application with a specified phone number without directly making the call.

Key Concepts:

Implicit Intent: Used when the target component is not known.

ACTION_DIAL: Opens the dialer with a phone number.

URI (tel): Phone number is passed using tel: URI scheme.

No Permission Required: ACTION_DIAL does not require CALL_PHONE permission.

Implicit intents help applications interact with other apps securely and efficiently.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	1
2.	<u>Android Studio</u>	<u>Latest Version</u>	1

J. Safety and necessary Precautions followed

1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

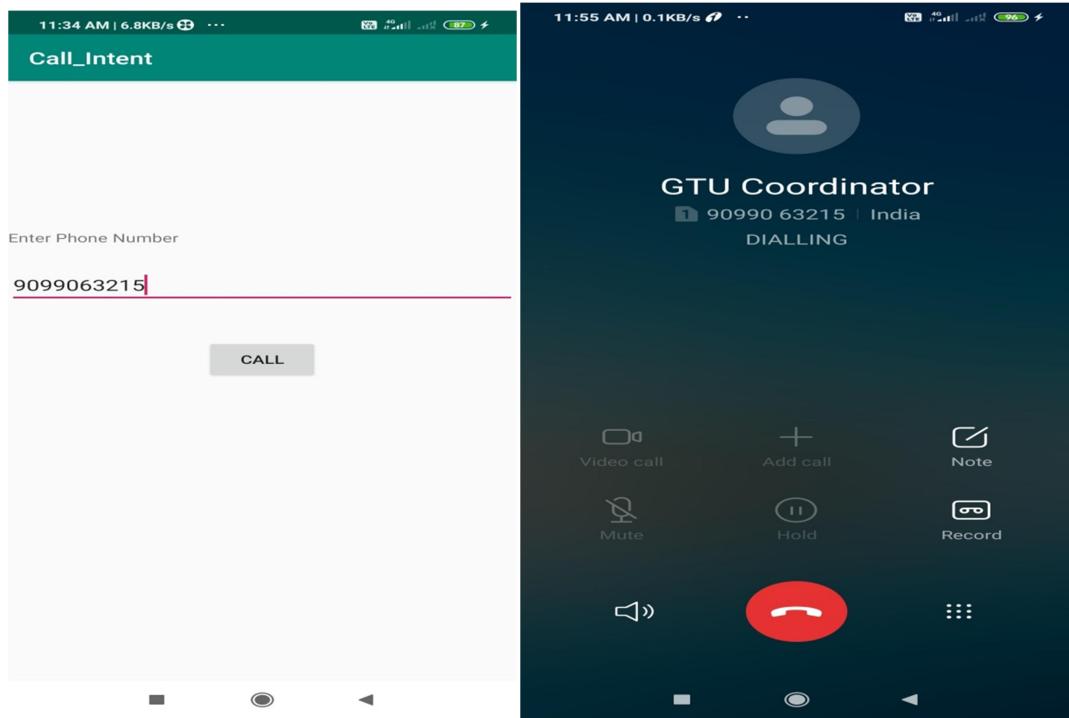
K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open **Android Studio** and create a new Android project.
2. Provide a suitable project name and select the minimum SDK.
3. Choose **Empty Activity** and click **Finish**.
4. Open `activity_main.xml`.
5. Add an `EditText` for phone number input and a `Button` to open the dialer.
6. Open `MainActivity.java`.
7. Write code to create an **Implicit Intent** using `ACTION_DIAL`.
8. Pass the phone number using `Uri.parse("tel:")`.

9. Run the application on Emulator or physical device.
10. Enter a phone number and click the button to open the dialer.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an implicit intent in Android?

2. What is the difference between ACTION_DIAL and ACTION_CALL?

3. Why does ACTION_DIAL not require permission?

4. What is the purpose of Uri.parse("tel:")?

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.8: Develop an Android application to change screen color as per the user choice from a menu.

A. Objective:

To develop an Android application that changes the background color of the screen based on the user's selection from a menu.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Use menu options to handle user interaction and dynamically update the user interface'

D. Expected Course Outcomes(Cos)

CO3: Develop interactive User Interfaces using layouts and views.

E. Practical Outcome(PRo)

Understand menu creation in Android and dynamically change screen background color based on user choice.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Android, a Menu is used to provide a set of options to the user. Menus allow users to perform actions such as changing settings or triggering events in the application.

Android supports different types of menus:

- Options Menu
- Context Menu

- Popup Menu

The Options Menu appears in the app bar (three-dot menu) and is commonly used for global actions such as changing screen color.

When a menu item is selected, the `onOptionsItemSelected()` method is invoked, allowing the application to respond to user actions such as changing the background color of a layout.

Key Concepts:

Menu Resource (menu.xml)
`onCreateOptionsMenu()`
`onOptionsItemSelected()`
`setBackgroundColor()`

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

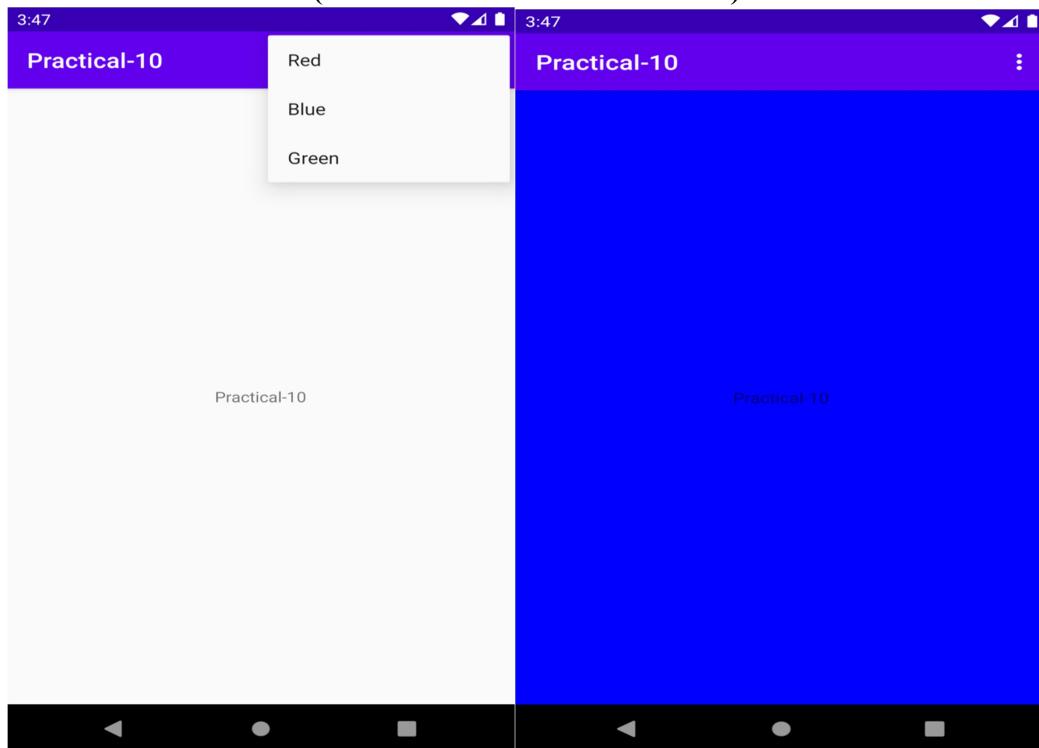
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open **Android Studio** and create a new Android project.
2. Provide a suitable project name and select the minimum SDK.
3. Choose **Empty Activity** and click **Finish**.
4. Open `activity_main.xml` and create a simple layout.
5. Create a menu resource file `menu_color.xml` inside `res/menu`.
6. Define menu items for different colors.
7. Open `MainActivity.java`.
8. Inflate the menu using `onCreateOptionsMenu()`.
9. Handle menu item clicks in `onOptionsItemSelected()`.
10. Change the background color of the layout based on selected menu option.
11. Run the application and test color changes.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an Options Menu in Android?

2. Which method is used to create a menu in Android?

3. How is a menu item click handled in Android?

4. Which method is used to change background color programmatically?

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.9: Develop an Android application that uses Content Providers to share data between different apps and components.

A. Objective:

To study and implement Content Providers in Android for sharing data between different applications using Content Resolver and URI.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: ‘Design Android applications that can securely share data between different applications’

D. Expected Course Outcomes(Cos)

CO2: Understand Android architecture and use its core components.

E. Practical Outcome(PRo)

After completing this practical, students will be able to:

- ✓ Create and register a **Content Provider** in an Android application.
- ✓ Share application data **securely** between different apps and components.
- ✓ Access shared data using **Content Resolver and Content URI**.
- ✓ Perform basic **CRUD operations** through Content Providers.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

What is a Content Provider?

A **Content Provider** is one of the main components of Android that is used to **share data between different applications and components**.

- ✓ It acts as an **interface** between an application's data and other applications.
- ✓ It allows access to data in a **controlled and secure manner**.
- ✓ Data is usually stored in an **SQLite database** and accessed through the Content Provider.

Example:

Contacts app, SMS app, Media gallery all use Content Providers.

Why is Content Provider Used?

- ✓ Android **does not allow direct database access** between applications.
- ✓ It provides a **standard and secure way** to share data.
- ✓ It controls **who can read or modify data**.
- ✓ It supports **structured data access**.
- ✓ It helps in **inter-application communication**.

Major Components of a Content Provider

When implementing a Content Provider in Android, the following **major components** must be understood:

1. Content Provider Class

- ✓ A Java class that **extends ContentProvider**.
- ✓ It acts as a **bridge between the database and other applications**.
- ✓ All data sharing logic is written inside this class.

2. SQLite Database

- ✓ Used to **store structured data internally**.
- ✓ Usually created using **SQLiteOpenHelper**.
- ✓ Content Provider accesses data from this database.

3. Authority

- ✓ A **unique name** that identifies the Content Provider.
- ✓ It is used in the **Content URI**.
- ✓ Must be unique across the device.

❖ **Example:** com.example.contentproviderdemo.provider

4. Content URI

- ✓ Used to **identify the data** provided by Content Provider.
- ✓ It specifies **which provider and which table** is accessed.

❖ **Format:** content://authority/table_name

5. Content Resolver

- ✓ Used by **client applications** to access data.
- ✓ It communicates with Content Provider using Content URI.
- ✓ Provides methods like `insert()`, `query()`, `update()`, and `delete()`.

6. UriMatcher

- ✓ Matches incoming Content URIs to predefined patterns.
- ✓ Helps the provider **decide which table or operation to perform**.
- ✓ Very important for **cross-application access**.

7. Cursor

- ✓ Used to **hold query results**.
- ✓ Returned by `query()` method.
- ✓ Client app reads data using Cursor.

8. AndroidManifest Registration

- ✓ Content Provider must be registered in `AndroidManifest.xml`.
- ✓ `android:exported="true"` is required for cross-app access.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

Part A: Provider App (Data Provider)

- ✓ Open **Android Studio**.
- ✓ Create a new project (Empty Activity)
 - **App Name:** ProviderApp
 - **Package:** com.example.contentproviderdemo
- ✓ Create a **SQLite database helper class** to store names.
- ✓ Create a class that **extends ContentProvider**.
 - Define **authority, Content URI**.
 - Implement required methods:
 - ◆ onCreate(), insert(), query(), update(), delete(), getType()
- ✓ Register the Content Provider in **AndroidManifest.xml** with
 - android:exported="true".
- ✓ Create a simple Activity to **insert sample names** into the database.
- ✓ Run the Provider App and insert data.

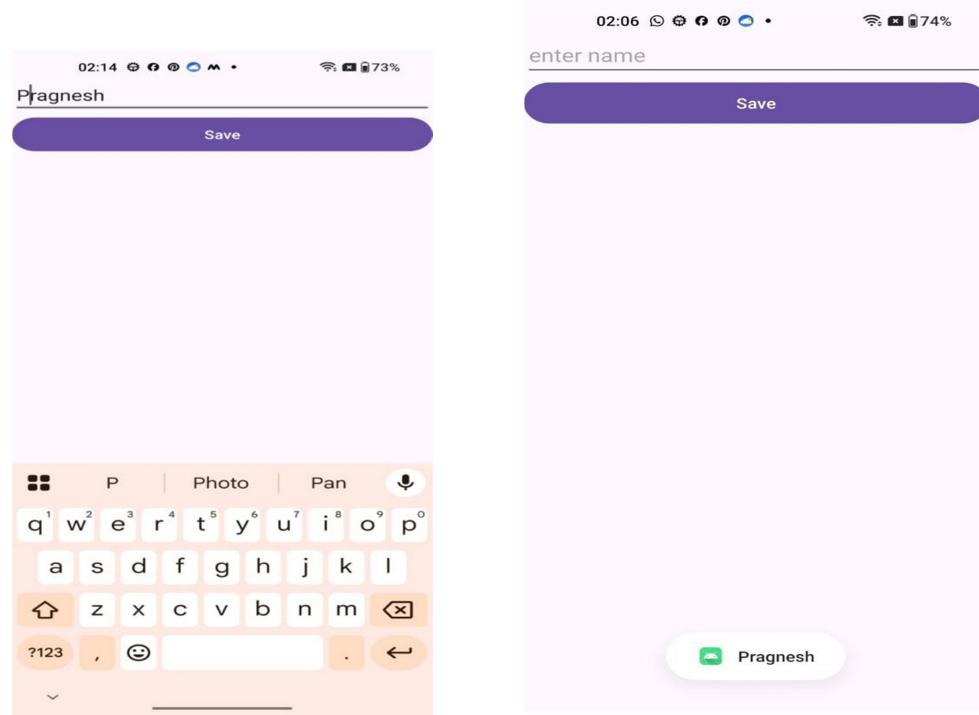
Part B: Client App (Data Consumer)

- ✓ Create another Android project (Empty Activity)
 - **App Name:** ClientApp
- ✓ Create the **same Content URI** using the provider's authority.
- ✓ Use **ContentResolver** to call `query()` on the Content URI.
- ✓ Read data using **Cursor**.
- ✓ Display retrieved names in a **TextView**.
- ✓ Run Provider App first, then run Client App.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):

Part A : Provider App



Part B : Client App



M. Practical related Quiz.

1. Why is Content Provider required in Android?

2. Name the major components of a Content Provider.

3. Give two real-life examples of applications that use Content Providers.

4. What is the role of **Content Resolver** in accessing a Content Provider?

5. What is a **Content URI**? Explain its format.

6. What is an **authority** in a Content Provider?

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.10: Develop an Android application that stores and retrieves username using shared preferences.

A. Objective:

To understand the concept of **Shared Preferences** in Android.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Design a simple Android application with local data storage (persistent storage) capability.'

D. Expected Course Outcomes(Cos)

CO4: Implement data storage using SQLite and Firebase Realtime Database.

E. Practical Outcome(PRo)

After completing this practical, students will be able to:

- ✓ Create an Android application that stores user input using Shared Preferences.
- ✓ Retrieve and display stored username correctly.
- ✓ Demonstrate understanding of local and persistent data storage in Android.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

What is Shared Preferences?

Shared Preferences is an Android storage mechanism used to store **small amounts of data** in the form of **key–value pairs**.

- ✓ It stores data **permanently**.
- ✓ The data remains saved even after the app is closed.

- ✓ It is mainly used to store **simple data**, not large files.

Examples of data stored:

- ✓ Username ,Password (not recommended for sensitive data),Login status,App settings

Why Use Shared Preferences?

- ✓ It is **easy to use**.
- ✓ It does **not require database creation**.
- ✓ It is suitable for storing **small and simple data**.
- ✓ Data is stored **locally inside the device**.
- ✓ Data remains available even after **app restart**.

How Shared Preferences Works?

Shared Preferences stores data in **key–value format**.

Key → unique name (example: "username")

Value → actual data (example: "Student123")

How to Use Shared Preferences (Store Data)

Steps to Store Username:

- ✓ Get the Shared Preferences object.
- ✓ Open the editor.
- ✓ Store data using key–value pair.
- ✓ Save the data.

Syntax (Java):

```
SharedPreferences sp = getSharedPreferences("MyPrefs", MODE_PRIVATE);
SharedPreferences.Editor editor = sp.edit();
editor.putString("username", "Student123");
editor.apply();
```

How to Use Shared Preferences (Retrieve Data)

Steps to Retrieve Username:

- ✓ Access the same Shared Preferences file.
- ✓ Use the key to get stored value.
- ✓ Provide default value if data is not found.

Syntax (Java):

```
SharedPreferences sp = getSharedPreferences("MyPrefs",  
MODE_PRIVATE);String uname = sp.getString("username", "No User Found");
```

Limitations of Shared Preferences

- ✓ Not suitable for large data
- ✓ Not secure for sensitive information
- ✓ Stores only primitive data types

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	1
2.	<u>Android Studio</u>	<u>Latest Version</u>	1

J. Safety and necessary Precautions followed

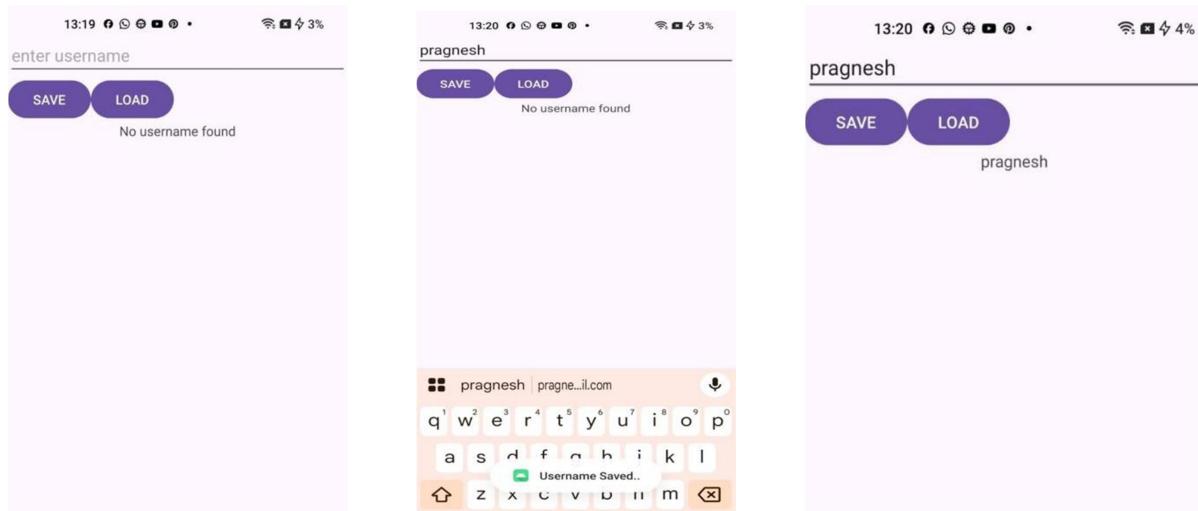
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open **Android Studio**.
2. Create a new project with **Empty Activity**.
3. Design UI in `activity_main.xml` with:
 - a) `EditText` for Username
 - b) Button to save Username
 - c) Button to Display Username
 - d) `Textview` for Username
4. Write Java code to:
 - a) Read username from `EditText`
 - b) Store it using Shared Preferences
5. Retrieve username from Shared Preferences in same activity with Load button activity.
6. Display username in `TextView`.
7. Run the application on emulator or physical device.
8. Verify that username is saved and displayed correctly.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



The screenshot shows the Android Studio interface with the Device Explorer tab selected. The left pane displays the XML code for `myprefDemo.xml`:

```
<?xml version='1.0' encoding='utf-8'>
<map>
    <string name="username">pragnesh</string>
</map>
```

The right pane shows the file system of the OnePlus IN2021 device. The `shared_prefs` directory under `com.example.mysharedpre` contains the `myprefDemo.xml` file, which is highlighted.

Name	Permissions	Date	Size
<code>com.coloros.weather.servic</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>com.cxinventor.file.explorei</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>com.example.mygeminiidem</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>com.example.mysharedpre</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>cache</code>	<code>drwxrws--x</code>	2025-12-16 13:18	3.4 KB
<code>code_cache</code>	<code>drwxrws--x</code>	2025-12-16 13:18	3.4 KB
<code>files</code>	<code>drwxrwx--x</code>	2025-12-16 13:18	3.4 KB
<code>shared_prefs</code>	<code>drwxrwx--x</code>	2025-12-16 13:20	3.4 KB
<code>myprefDemo.xml</code>	<code>-rw-rw----</code>	2025-12-16 13:20	116 B
<code>com.facebook.appmanager</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>com.facebook.katana</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB
<code>com.facebook.orca</code>	<code>drwxrwx--x</code>	2024-01-01 10:30	4 KB

M. Practical related Quiz.

1. What is Shared Preferences used for?
2. In which format data is stored in Shared Preferences?
3. Which method is used to store data in Shared Preferences?
4. Which method is used to retrieve data from Shared Preferences?
5. Does Shared Preferences require any permission?

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.html>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:	
Criteria	Max Marks	Rubrics	Marks	
C1: Program Correctness	5	Excellent (5 marks): Completed programs correctly as per the requirements.		
		Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.		
		Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.		
		Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.		
C2: Coding methodology	5	Excellent (5 marks): The code is clean, well-organized, and very easy to understand.		
		Adequate (3-4 marks): The code is easy to understand.		
		Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.		
		Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.		
C3: User Interface (UI) Design	5	Excellent (5 marks): Easy to use and attractive User Interface.		
		Adequate (3-4 marks): Easy-to-use User Interface.		
		Poor (1-2 marks): Poor User Interface design.		
		Unsatisfactory (0 marks): Very difficult User Interface to use by the user.		
C4: Testing and Debugging	5	Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.		
		Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.		
		Poor (1-2 marks): Poor testing and debugging knowledge.		
		Unsatisfactory (0 marks): Not able to test and debug a program.		
C5: Submission in time	5	Excellent (5 marks): Submitted within the time limit.		
		Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.		
		Poor (1-2 marks): Submitted within a week after a given time limit.		
		Unsatisfactory (0 marks): Submitted after a week over the given time limit.		
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5				
Signature with Date:				

Date:

Practical No.11: Develop an android application that creates a database using SQLiteOpenHelper Class and performs Insert and Read from the SQLite database.

A. Objective:

To understand SQLiteOpenHelper class for database operations in Mobile Application development.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*
5. **Life-long learning:** *Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Insert and read the data from SQLite database using SQLiteOpenHelper class.'

D. Expected Course Outcomes(COs)

CO4: Implement data storage using SOLite and Firebase Realtime Database.

E. Practical Outcome(PRo)

Insert and read data from SQLite database using SQLiteOpenHelper class in Mobile application.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Mobile application development, SQLitedatabase is used to store the data in the device. SQLiteOpenHelper class is used to perform insert, read, update and delete the data from the SQLite database.

Key Concepts:

- The android.database.sqlite.SQLiteOpenHelper class is used for database creation and version management.
- For performing any database operation, you have to provide the implementation of **onCreate()** and **onUpgrade()** methods of SQLiteOpenHelper class.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No •	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

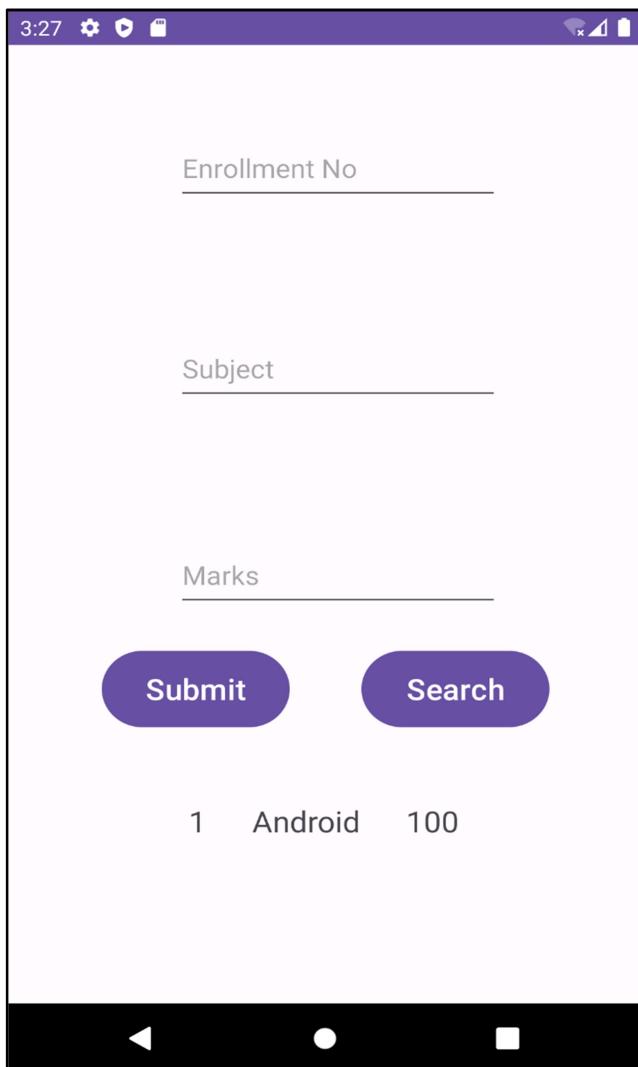
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Add EditText to insert the data to SQLite database.
6. Create a java class which extends the SQLiteOpenHelper class.
7. Perform insert and read operations using SQLiteDatabase methods.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an SQLite database?

2. Write a query to insert the data in SQLite database.

3. Write a query to read the data from SQLite database.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:
Criteria	Max Marks	Rubrics	Marks
C1: Program Correctness	5	<p>Excellent (5 marks): Completed programs correctly as per the requirements.</p> <p>Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.</p> <p>Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.</p> <p>Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.</p>	
C2: Coding methodology	5	<p>Excellent (5 marks): The code is clean, well-organized, and very easy to understand.</p> <p>Adequate (3-4 marks): The code is easy to understand.</p> <p>Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.</p> <p>Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.</p>	
C3: User Interface (UI) Design	5	<p>Excellent (5 marks): Easy to use and attractive User Interface.</p> <p>Adequate (3-4 marks): Easy-to-use User Interface.</p> <p>Poor (1-2 marks): Poor User Interface design.</p> <p>Unsatisfactory (0 marks): Very difficult User Interface to use by the user.</p>	
C4: Testing and Debugging	5	<p>Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.</p> <p>Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.</p> <p>Poor (1-2 marks): Poor testing and debugging knowledge.</p> <p>Unsatisfactory (0 marks): Not able to test and debug a program.</p>	
C5: Submission in time	5	<p>Excellent (5 marks): Submitted within the time limit.</p> <p>Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.</p> <p>Poor (1-2 marks): Submitted within a week after a given time limit.</p> <p>Unsatisfactory (0 marks): Submitted after a week over the given time limit.</p>	
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5			
Signature with Date:			

Date:

Practical No.12: Develop an android application to Update and Delete data from the SQLite database using SQLiteOpenHelper class.

A. Objective:

To understand SQLiteOpenHelper class for database operations in Mobile Application development.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*
5. **Life-long learning:** *Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: ‘Update and delete the data from the SQLite database using SQLiteOpenHelper class.’

D. Expected Course Outcomes(Cos)

CO4: Implement data storage using SQLite and Firebase Realtime Database.

E. Practical Outcome(Pro)

Update and Delete data from SQLite database using SQLiteOpenHelper class in Mobile application.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

In Mobile application development, SQLite database is used to store the data in the device. SQLiteOpenHelper class is used to perform insert, read, update and delete the data from the SQLite database.

Key Concepts:

- The android.database.sqlite.SQLiteOpenHelper class is used for database creation and version management.
- For performing any database operation, you have to provide the implementation of **onCreate()** and **onUpgrade()** methods of SQLiteOpenHelper class.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	1
2.	<u>Android Studio</u>	<u>Latest Version</u>	1

J. Safety and necessary Precautions followed

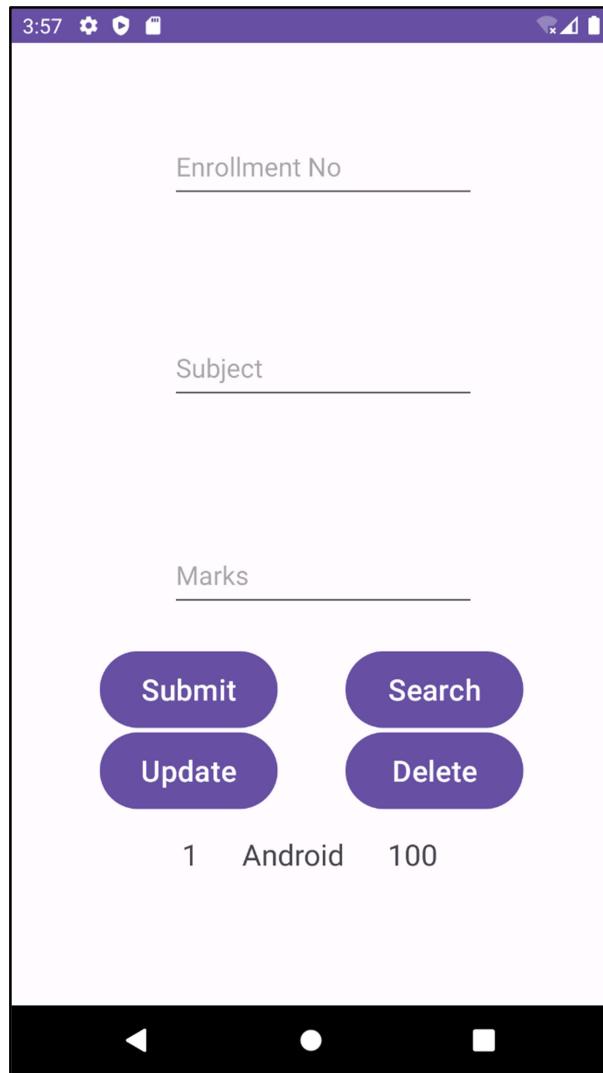
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Add EditText to insert the data to SQLite database.
6. Create a java class which extends the SQLiteOpenHelper class.
7. Perform update and delete operations using SQLiteDatabase methods.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an SQLite database?
 2. Write a query to update the data in SQLite database.
 3. Write a query to delete the data from SQLite database.

N. References / Suggestions

1. <https://developer.android.com>
 2. <https://www.javatpoint.com/android-tutorial>
 3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:
Criteria	Max Marks	Rubrics	Mark s
C1: Program Correctness	5	<p>Excellent (5 marks): Completed programs correctly as per the requirements.</p> <p>Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.</p> <p>Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.</p> <p>Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.</p>	
C2: Coding methodology	5	<p>Excellent (5 marks): The code is clean, well-organized, and very easy to understand.</p> <p>Adequate (3-4 marks): The code is easy to understand.</p> <p>Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.</p> <p>Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.</p>	
C3: User Interface (UI) Design	5	<p>Excellent (5 marks): Easy to use and attractive User Interface.</p> <p>Adequate (3-4 marks): Easy-to-use User Interface.</p> <p>Poor (1-2 marks): Poor User Interface design.</p> <p>Unsatisfactory (0 marks): Very difficult User Interface to use by the user.</p>	
C4: Testing and Debugging	5	<p>Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.</p> <p>Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.</p> <p>Poor (1-2 marks): Poor testing and debugging knowledge.</p> <p>Unsatisfactory (0 marks): Not able to test and debug a program.</p>	
C5: Submission in time	5	<p>Excellent (5 marks): Submitted within the time limit.</p> <p>Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.</p> <p>Poor (1-2 marks): Submitted within a week after a given time limit.</p> <p>Unsatisfactory (0 marks): Submitted after a week over the given time limit.</p>	
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5			
Signature with Date:			

Date:

Practical No.13: Develop an Android application that integrates Firebase real-time database and store the data into it.

A. Objective:

To understand Firebase database for Mobile Application development.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*
5. **Life-long learning:** *Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: 'Integrate the Firebase database to mobile application and perform insertion of the data.'

D. Expected Course Outcomes(Cos)

CO4: Implement data storage using SQLite and Firebase Realtime Database.

E. Practical Outcome(PRo)

Understand Firebase integration to the mobile application and perform insertion of the data in the firebase database.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

The Firebase Realtime Database is a cloud-hosted database. Data is stored in JSON format and synchronized in real time to every connected application. When you

build mobile application you can use firebase database to Create Read, Update, and Delete the database from your application.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	1
2.	<u>Android Studio</u>	<u>Latest Version</u>	1

J. Safety and necessary Precautions followed

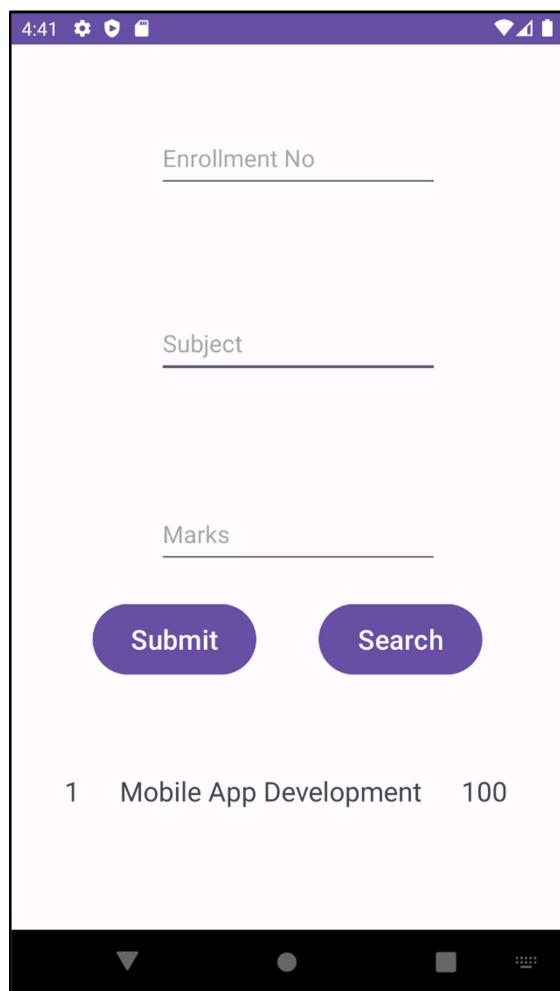
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. Open “Tools” menu and select firebase.
6. Select firebase realtime database from the firebase assistant pane.
7. Click on connect to firebase.
8. Create a project in firebase console.
9. Add Firebase SDK to your app.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an Firebase realtime database?

2. Write steps to integrate firebase realtime database in your application.

3. Write the benefit of Firebase realtime database.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:
Criteria	Max Marks	Rubrics	Mark s
C1: Program Correctness	5	<p>Excellent (5 marks): Completed programs correctly as per the requirements.</p> <p>Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.</p> <p>Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.</p> <p>Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.</p>	
C2: Coding methodology	5	<p>Excellent (5 marks): The code is clean, well-organized, and very easy to understand.</p> <p>Adequate (3-4 marks): The code is easy to understand.</p> <p>Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.</p> <p>Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.</p>	
C3: User Interface (UI) Design	5	<p>Excellent (5 marks): Easy to use and attractive User Interface.</p> <p>Adequate (3-4 marks): Easy-to-use User Interface.</p> <p>Poor (1-2 marks): Poor User Interface design.</p> <p>Unsatisfactory (0 marks): Very difficult User Interface to use by the user.</p>	
C4: Testing and Debugging	5	<p>Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.</p> <p>Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.</p> <p>Poor (1-2 marks): Poor testing and debugging knowledge.</p> <p>Unsatisfactory (0 marks): Not able to test and debug a program.</p>	
C5: Submission in time	5	<p>Excellent (5 marks): Submitted within the time limit.</p> <p>Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.</p> <p>Poor (1-2 marks): Submitted within a week after a given time limit.</p> <p>Unsatisfactory (0 marks): Submitted after a week over the given time limit.</p>	
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5			
Signature with Date:			

Date:

Practical No.14: Develop an Android application to retrieve data from Firebase Realtime Database or a JSON data source (REST API), parse the JSON response, and display the data in a Recycler View.

A. Objective:

To understand Firebase database for Mobile Application development.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** *Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.*
2. **Problem analysis:** *Identify and analyse well-defined engineering problems using codified standard methods.*
3. **Design/ development of solutions:** *Design solutions for engineering well-defined technical problems and assist with the design of systems components or processes to meet specified needs.*
4. **Engineering Tools, Experimentation and Testing:** *Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.*
5. **Life-long learning:** *Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.*

C. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry-identified competency: ‘Retrieve the data from the firebase database or JSON/REST API and display it in the Android application’

D. Expected Course Outcomes(Cos)

CO5: Explore and demonstrate modern technologies such as Kotlin, Flutter (Cross-platform development), APIs, and JSON parsing.

E. Practical Outcome(PRo)

Retrieve the data from the Firebase database or a JSON data source (REST API) and display it in the Android Application using RecyclerView.

F. Expected Affective domain Outcome(ADos)

1. Handle computer systems carefully with safety and necessary precaution
2. Turn off systems after completion of practical lab to save power.

G. Prerequisite Theory:

The Firebase Realtime Database is a cloud-hosted database. Data is stored in JSON format and synchronized in real time to every connected application. When you build mobile application you can use firebase database to Create Read, Update, and Delete the database from your application. You can also build mobile application where you can use REST API to read, update, delete the data from the database.

H. Experimental set up/ Program Logic-Flow chart:

I. Resources/Equipment Required

Sr.No •	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
<u>1</u>	<u>Computer system</u>	<u>8 GB RAM, intel core-i5 processor and 128 GB SSD(recommended).</u>	<u>1</u>
<u>2.</u>	<u>Android Studio</u>	<u>Latest Version</u>	<u>1</u>

J. Safety and necessary Precautions followed

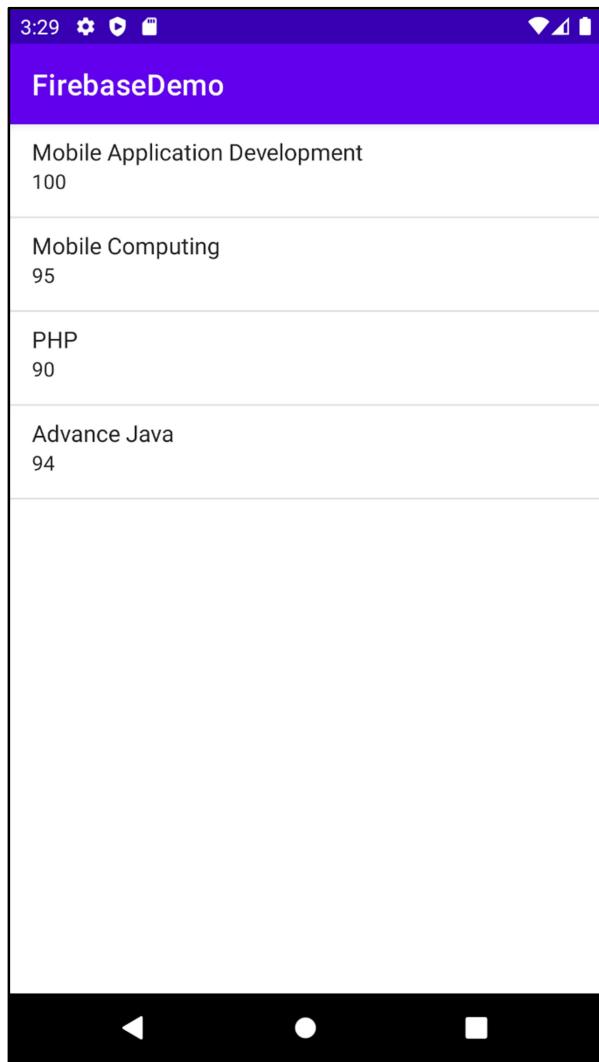
1. Turn off the power switch only after the computer is shut down.
2. Do not plug out any computer cables.
3. Save your work/program periodically.

K. Procedure to be followed/Source code (CE & IT software subjects):

1. Open Android Studio and create a new Android project.
2. Provide a suitable name for the project and select the desired SDK version and minimum SDK level.
3. Choose an Empty Activity template and click "Finish" to create the project.
4. Android Studio will generate the necessary project structure and code.
5. If you are using Firebase, integrate Firebase to your app if not done.
6. Create a RecyclerView and Adapter to display the data.
7. Analyze the JSON data that you are going to retrieve from Firebase or API.
8. Retrieve the data from the Firebase or any REST API and display in the RecyclerView.

L. Observations and Calculations/Input-Output (CE & IT software subjects):

SAMPLE OUTPUT (PREVIEW ON MOBILE/AVD):



M. Practical related Quiz.

1. What is an Firebase realtime database?

2. Write steps to integrate firebase realtime database in your application.

3. Define JSON data format.

N. References / Suggestions

1. <https://developer.android.com>
2. <https://www.javatpoint.com/android-tutorial>
3. <https://www.tutorialspoint.com/android/index.htm>

O. Assessment-Rubrics

Practical:	Name of Student:		E.No.:
Criteria	Max Marks	Rubrics	Marks
C1: Program Correctness	5	<p>Excellent (5 marks): Completed programs correctly as per the requirements.</p> <p>Adequate (3-4 marks): Completed programs correctly with approx. 50% - 70% requirements.</p> <p>Poor (1-2 marks): Completed programs correctly with 30% - 50% requirements.</p> <p>Unsatisfactory (0 marks): Completed programs correctly with less than 30% requirements.</p>	
C2: Coding methodology	5	<p>Excellent (5 marks): The code is clean, well-organized, and very easy to understand.</p> <p>Adequate (3-4 marks): The code is easy to understand.</p> <p>Poor (1-2 marks): The code is readable only by someone who knows what it is supposed to be doing.</p> <p>Unsatisfactory (0 marks): The code is poorly organized and very difficult to understand.</p>	
C3: User Interface (UI) Design	5	<p>Excellent (5 marks): Easy to use and attractive User Interface.</p> <p>Adequate (3-4 marks): Easy-to-use User Interface.</p> <p>Poor (1-2 marks): Poor User Interface design.</p> <p>Unsatisfactory (0 marks): Very difficult User Interface to use by the user.</p>	
C4: Testing and Debugging	5	<p>Excellent (5 marks): Tested the program with all the test cases and debugged it to find all the bugs.</p> <p>Adequate (3-4 marks): Tested with adequate test cases for the program to run and debugged it.</p> <p>Poor (1-2 marks): Poor testing and debugging knowledge.</p> <p>Unsatisfactory (0 marks): Not able to test and debug a program.</p>	
C5: Submission in time	5	<p>Excellent (5 marks): Submitted within the time limit.</p> <p>Adequate (3-4 marks): Submitted within One or Two Days after the given time limit.</p> <p>Poor (1-2 marks): Submitted within a week after a given time limit.</p> <p>Unsatisfactory (0 marks): Submitted after a week over the given time limit.</p>	
Total Marks for Practical = Marks of C1 + C2 + C3 + C4 + C5			
Signature with Date:			

Mobile Application Development

DI04016051

Lab manuals are prepared by

Shri Roshankumar R. Rohit

Lecturer

Government Polytechnic for Girls, Surat

Shri Pragneshkumar A. Patel

Lecturer

Government Polytechnic for Girls, Surat

Shri Chinkit D. Suthar

Lecturer

Government Polytechnic, Himmatnagar

Branch Coordinator

Committee Chairman