LAB MANUAL

STUDENTS VERSION

SMART DEVICE PROGRAMMING LAB

(SIXTH SEMESTER, COURSE CODE: 6138)

Department Of Computer Engineering
Government Polytechnic College, Perumbavoor
KoovappadyP.O,Phone:04842649251,email:gptcpbvr@gmail.com

INDEX

Sl No:	Name of Experiment	Date	Remarks
	VISION AND MISSION		
	PEO, PO AND PSOS OF THE PROGRAM		
	GENERAL INSTRUCTIONS		
1	SAFETY PROCEDURES		
2	FAMILIARIZATION OF ANDROID STUDIO		
3	INSTALLING ANDROID STUDIO ON UBUNTU		
4	INSTALLING ANDROID STUDIO ON WINDOWS		
5	CREATING ANDROID VIRTUAL DEVICE		
6	HELLO WORLD APPLICATION		
7	TOAST A MESSAGE ON BUTTON CLICK		
8	LIFE CYCLE OF AN ANDROID ACTIVITY		
9	PERFORM ADDITION AND TOAST SUM OF BUTTON CLICK		
10	UNDERSTANDING TOGGLE BUTTON		
11	UNDERSTANDING CHECKBOX		
12	UNDERSTANDING RADIO GROUP		
13	EXPLICIT INTENT		
14	EXPLICIT INTENT – RADIO BUTTON		
15	EXPLICIT INTENT – CHECKBOX		
16	RESET BUTTON		
17	MAKE A PHONE CALL		
18	SQLITE-STUDENT DATABASE		

General Remarks (*for office use only*)							
Test 1:		Test 2:		Assign 1:		Assign 2:	

VISION AND MISSION

Government Polytechnic College, Perumbavoor
Vision: Excel as a centre of skill education moulding professionals who sincerely strive for the betterment of society **Mission:** • To impart state of the art knowledge and skill to the graduate and moulding them to be competent, committed and responsible for the well-being of society • To apply technology in the traditional skills, thereby enhancing the living standard of the community

Department of Computer Engineering
Vision: Excel as a center of skill education in Computer Engineering moulding professionals who sincerely strive for the betterment of themselves and the society. **Mission:** • To impart state of the art knowledge, skill and attitude to the graduates and contribute to their sustainable development • To merge technologies in the field of computer engineering with occupational skills, thereby improving the quality of living

PEO, PO AND PSOs OF THE PROGRAM

PROGRAM OUTCOMES

PO1: Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

PO2: Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

PO3: Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

PO4: Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

PO5: Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.

PO6: 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.

PO7: Life-long learning: Ability to analyse individual needs and engage in updating in the context of technological changes.

PROGRAM SPECIFIC OUTCOMES (PSOS)

PSO1: Apply concepts and knowledge in the field of software systems, hardware and networking with concern for the society.

PSO2: Generate ideas from the knowledge of engineering specialization leading to professional growth.

PSO3: Apply knowledge and understanding of engineering principles to initiate entrepreneurship ventures.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO1: Secure successful careers in hardware and software design, development, testing, maintenance and marketing.

PEO2: Acquire knowledge and competency in the domain to develop innovative, cost effective and socially acceptable solutions to engineering problems in a multi-disciplinary work environment.

PEO3: Develop strong fundamental knowledge that prepares them for professional careers/ higher studies with attitude for lifelong learning.

PEO4: Instill the attitude to be sensitive to ethical, societal and environmental issues while pursuing their professional duties.

PEO5: Possess leadership qualities and be effective communicator to work efficiently with diverse teams, promote and practice appropriate ethical practices.

GENERAL INSTRUCTIONS

Rough record and Fair record are needed to record the experiments conducted in the laboratory. Rough records are needed to be certified immediately on completion of the experiment. Fair records are due at the beginning of the next lab period. Fair records must be submitted as neat, legible, and complete.

INSTRUCTIONS TO STUDENTS FOR WRITING THE FAIR RECORD

In the fair record, the index page should be filled properly by writing the corresponding experiment number, experiment name , date on which it was done and the page number.

On the right side page of the record following has to be written:

1. Title: The title of the experiment should be written in the page in capital letters.

2.Exp No: And Date: In the top margin, experiment number and date should be written.

3. Aim: The purpose of the experiment should be written clearly.

4.Principle/Theory: Simple algorithm should be written

5. Procedure: Steps for doing the experiment.

6. Program: Simple working of the algorithm should be written.

7. Results: The results of the experiment must be summarized in writing and should be fulfilling the aim.

On the Left side page of the record following has to be recorded:

1. **Input:** Input of the program given
2. **Output :** Output of the program
3. **Design:** The design of the output (if necessary).

Exp No : 01 **Date :** D D / M M / Y Y

SAFETY PROCEDURES

Problem Statement:

The safety instructions are presented to the attention of the students as a mean of preventing accidents while performing experiments and activities in Software lab of the department .The purpose is to draw attention to the risks involved in lab activities to prevent human suffering and damage to equipment.

Safety in the laboratory:

Working in the lab is not allowed without following electricity precautions displayed.

No individual work is allowed in the lab.

Laboratory in charge is responsible for the arrangements of your lab activities; Listen carefully to his/her instructions and follow them.

To do and not to do:

Inform the lab in charge about dangerous conditions and faults in the lab or nearby environment.

Do not do any action that may harm people or equipment in the lab.

Do not misuse any of the tools or instruments belong to the lab.

Strict discipline should be maintained in the laboratory.

Turn off cell phones before entering the lab.

At the end and beginning of laboratory, follow 5S procedures and leave the work table clean and tidy.

Electrical Safety:

Consult Electrical Engineering section available in the campus for electrical safety queries.

The lab equipment is powered from electrical sockets installed on the tables.

Do not use equipment that is powered from a damaged socket.

Do not use equipment that is powered from flexible cable with damaged insulation or if it's plug is not assembled properly.

Do not repair or disassemble electrical equipment including replacement of fuses installed in the equipment.

Do not open the main fuse box, unless it is an emergency and you need to switch off main circuit breaker.

Be sure to turn off the power and remove the power plug from all equipment before working repairing or assembling.

Do not plug in or remove equipment while the power is on.

Emergency Switches:

The laboratory has circuit breakers, which is located in the main panel. Identify the place.

In an emergency condition, switch off circuit breakers immediately.

Result:

Familiarization of safety precautions performed

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 02 **Date :** D D / M M / Y Y

FAMILIARIZATION OF ANDROID STUDIO

Android is a software package and Linux based operating system for mobile devices such as tablet computers and smartphones. It is developed by Google and later the OHA (Open Handset Alliance). It's a consortium of 84 companies such as google, samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc.It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Plateform.Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users.There are many code names of android such as Lollipop, Kitkat,Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut etc

Android Core Building Blocks

An android component is simply a piece of code that has a well defined life cycle e.g. Activity, Receiver, Service etc. The core building blocks or fundamental components of android are activities, views, intents, services, content providers, fragments and AndroidManifest.xml.

- **Activity**-An activity is a class that represents a single screen. It is like a Frame in AWT.
- **View**-A view is the UI element such as button, label, text field etc. Anything that you see is a view.
- **Intent**-Intent is used to invoke components.
- **Service**-Service is a background process that can run for a long time. There are two types of services local and remote. Local service is accessed from within the application whereas remote service is accessed remotely from other applications running on the same device.
- **Content Provider**-Content Providers are used to share data between the applications.
- Fragment-Fragments are like parts of activity. An activity can display one or more fragments on the screen at the same time.

- **AndroidManifest.xml**-It contains information about activities, content providers, permissions etc. It is like the web.xml file in Java EE.
- **Android Virtual Device (AVD)**-It is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.

Android Emulator

Android Emulator is used to run, debug and test the android application. If you don't have the real device, it can be the best way to run, debug and test the application. It uses an open source processor emulator technology called **QEMU**. The emulator tool enables you to start the emulator from the command line.

OUTPUT

RESULT

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 03 **Date :** D D / M M / Y Y

INSTALLING ANDROID STUDIO ON UBUNTU

With the advancement of smart phones in the recent years, Android has become one of the biggest phone platforms and all the tools required to build Android applications are also freely available. Android Studio is an Integrated Development Environment (IDE) for developing Android applications. Installation on Ubuntu 15.04 We can install Android Studio in two ways.

- The first method is to set up the required repository and install it,
- The second method is to download it from the official Android site and install it locally.

In the first method, we will be setting up the repo using command line and install it. Before proceeding, we need to make sure that we have JDK version1.6 or greater installed.

To install JDK 1.8.

Step1 : $ sudo add-apt-repository ppa:webupd8team/java

Step2: $ sudo apt-get up

Step 3: $ sudo apt-get install oracle-java8-installer oracle-java8-set-default Verify if java installation was successful:

Step4: :~$ java -version

To install Android Studio

Step1: Setup the repo for installing Android Studio

$ sudo apt-add-repository ppa:paolorotolo/android-studio

Step2: $ sudo apt-get up

Step3: $ sudo apt-get install android-studio

Above install command will install android-studio in the directory /opt. Now, run the following command to start the setup wizard:

Step4: $ /opt/android-studio/bin/studio.sh

This will invoke the setup screen.

Step5:Click Next->Next. Once you press the Finish button, Licence agreement will be displayed. After you accept the licence, it starts downloading the required components.

Android studio installation will be complete after this step. When you relaunch Android studio, you will be shown the following welcome screen from where you will be able to start working with your Android Studio.

OUTPUT

RESULT

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 04 **Date :** D D / M M / Y Y

INSTALLING ANDROID STUDIO ON WINDOWS

Android Studio is one of the best and most popular IDE for development of Android application which is released by IntelliJ IDEA. Android studio has rich functionalists and easy to work. Android studio has great and awesome user interface, it is also smart with coding and will save time. To download Android Studio, it is need to visit official site of android. There will be android studio download button. After downloading android studio, follow below steps to install android studio on your computer.

Step1: Open you downloaded android studio file and wait for few seconds, it will take some time to load on your screen.

Step2: After that the window will appear on your screen, where it will show you welcome to android studio. Go for "Next"

Step3: Now choose what you want to install, all are most important component in android programming. Go for "Next"

Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation.

Step5: Go for "I Agree", It's for Haxm installation.

Step6: Now choose the location for Android Studio and SDK. Go for "Next"

Step7: Here we need to define the size of android emulator processor. We can define up to 2GB and more than that, depends on your RAM capacity. The size will be determined through Haxm. This will help android studio to run faster.

Step8: Now choose setting for shortcut of android studio on your computer.

Step9: After that it will take lots of time to copy all the files to the storage location you have selected before.

OUTPUT

RESULT

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 05 **Date :** D D / M M / Y Y

CREATING ANDROID VIRTUAL DEVICE

For creating the new android studio project:

1. In the **Welcome to Android Studio** window, click **Start a new Android Studio project**. or if you have a project opened, select **File > New Project**.
2. In the **Create New Project** window, enter the following values:

   Application Name: "My First App"

   Company Domain: "example.com"

You might want to change the project location. Also, if you want to write a Kotlin app, check the **Include Kotlin support** checkbox. Leave the other options as they are.

3.Click **Next**.

4.In the **Target Android Devices** screen, keep the default values and click **Next**.

5.In the **Add an Activity to Mobile** screen, select **Empty Activity** and click **Next**.

6.In the **Configure Activity** screen, keep the default values and click **Finish**.

After some processing, Android Studio opens the IDE.

Run the app

Run the app on an emulator as follows:

1.In Android Studio, click the **app** module in the **Project** window and then select **Run > Run**

2.In the **Select Deployment Target** window, click **Create New Virtual Device**.

3.In the **Select Hardware** screen, select a phone device, such as Pixel, and then click **Next**.

4.In the **System Image** screen, select the version with the highest API level. If you don't have that version installed, a **Download** link is shown, so click that and complete the download.

5.Click **Next**.

6.On the **Android Virtual Device (AVD)** screen, leave all the settings alone and click **Finish**.

7.Back in the **Select Deployment Target** dialog, select the device you just created and click **OK**.

Android Studio installs the app on the emulator and starts it,"Hello World!" displayed in the app running on the emulator.

OUTPUT

RESULT

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 06 **Date :** D D / M M / Y Y

HELLO WORLD APPLICATION

AIM:

To create an application and display “Hello World”

PROCEDURE

1. In Android studio, create new project.
2. Fill out the fields on the screen and click Next.
3. Click next and select blank activity and the click Next.
4. Under the Customize the Activity, change the Activity Name and Layout Name.
5. Click the Finish button to create the project
6. Run the app on emulator by choosing an android virtual device.

PROGRAM

MainActivity.java

/**********Write the java code**********/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here**********/**

RESULT

/*****************Write the result***************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 07 **Date :** D D / M M / Y Y

TOAST A MESSAGE ON BUTTON CLICK

AIM:

To create an android application to understand how to display notification on button click

THEORY

Android Toast can be used to display information for the short period of time. A toast contains message to be displayed quickly and disappears after sometime.

Methods of Toast class

Method	Description
makeText(Context context, CharSequence text, int duration)	makes the toast containing text and duration.
public void show()	displays toast.

PROCEDURE

1. **Create a new project**
2. **Enter Application details.**
3. In .xml file create a button view and set the attribute : android:text="Display toast message"
4. In java file
    a. Create objects of Button and Toast class
    b. Set a click listener on the button object
    c. Specify the message to toast inside onClick() method
    d. Invoke show() method of Toast class to display the notification

5. Execute the application by clicking the menu Run -> Run

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display Android activity*******/

OUTPUT

/****Draw the obtained output here**********/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 08 **Date :** D D / M M / Y Y

LIFE CYCLE OF AN ANDROID ACTIVITY

AIM:

To understand different methods that control the life cycle of an activity.

THEORY

Activity is one of the building blocks of Android OS. In simple words Activity is a screen that user interact with. Every Activity in android has lifecycle like created, started, resumed, paused, stopped or destroyed. These different states are known as Activity Lifecycle. Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class.

onCreate() – Called when the activity is first created

onStart() – Called just after it's creation or by restart method after onStop(). Here Activity start becoming visible to user

onResume() – Called when Activity is visible to user and user can interact with it

onPause() – Called when Activity content is not visible because user resume previous activity

onStop()–Called when activity is not visible to user because some other activity takes place of it

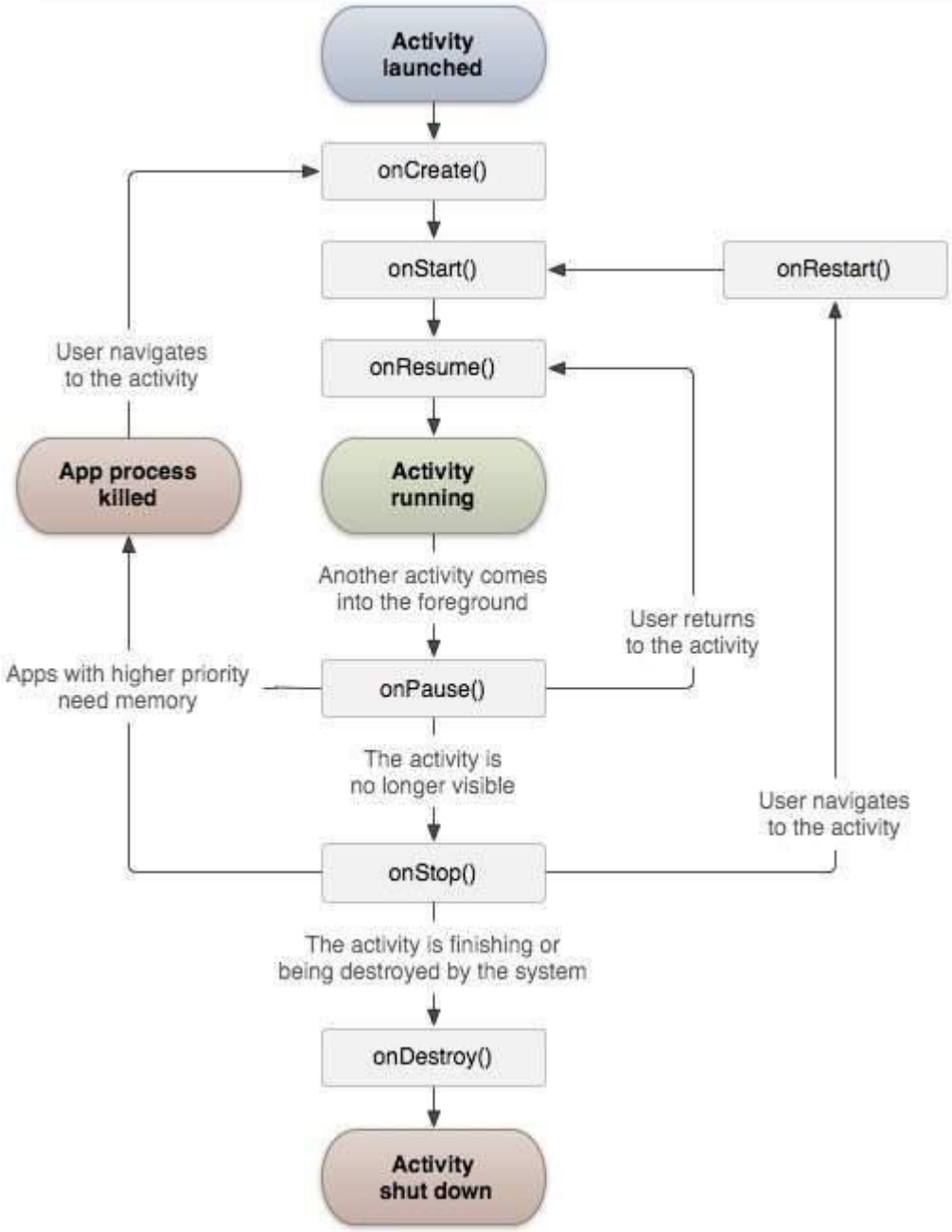
onRestart() – Called when user comes on screen or resume the activity which was stopped

onDestroy – Called when Activity is not in background

PROCEDURE

1. Create a new Android Project
2. Enter Application details
3. In java file, override all activity lifecycle method in Android and use Log class to print the message in Logcat.

4. Execute the application by clicking the menu Run -> Run
5. Go to Logcat present inside Android Monitor



Life cycle of an Activity

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display Android activity*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 09 **Date :** D D / M M / Y Y

TOAST SUM OF TWO NUMBERS ON BUTTON CLICK

AIM:

To familiarize UI widget Text field.

THEORY

The **EditText** is the standard text entry widget in Android apps. If the user needs to enter text into an app, this is the primary way for them to do that. Getting the value of the text entered into an EditText is as follows:

```
EditText simpleEditText = (EditText) findViewById(R.id.et_simple);
String strValue = simpleEditText.getText().toString();
```

PROCEDURE

1. Create a new Android Project
2. Enter Application details
3. Create two EditText instances and a button by declaring it inside XML file
4. In java file

a. Create objects of Button and TextField class

b. Set a click listener on the button object

c. Retrieve the values from two TextFields.

d. Specify the addition operation inside onClick() method

e. Invoke show() method of Toast class to display the sum.

5. Execute the application by clicking the menu Run -> Run

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here**********/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 10 **Date :** D D / M M / Y Y

UNDERSTANDING TOGGLE BUTTON

AIM

To familiarize toggle button and its attributes.

THEORY

Android Toggle Button can be used to display checked/unchecked (On/Off) state on the button.It is beneficial if user have to change the setting between two states. It can be used to On/Off Sound, Wifi, Bluetooth etc.

XML Attribute

- android:textOff- The text for the button when it is not checked
- android:textOn- The text for the button when it is checked.

Methods of ToggleButton class

- getTextOff()-Returns the text when button is not in the checked state.
- getTextOn()-Returns the text for when button is in the checked state.
- setChecked(boolean checked)- Changes the checked state of this button.

PROCEDURE

1. Create a new Android Project
2. **Enter Application details**
3. Create two ToggleButton instances and a button by declaring it inside xml file
4. In java file
    a. Create objects of Button and ToggleButton class
    b. Set a click listener on the button object
    c. initiate a toggle button
    d. check current state of a toggle button (true or false) by using isChecked() method .
    e. Display the text of current state of ToggleButtons using a Toast.
5. **Execute the application by clicking the menu Run -> Run**

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 11 **Date :** D D / M M / Y Y

UNDERSTANDING CHECKBOX

AIM:

To familiarize attributes and methods of CheckBox class.

THEORY

In Android, CheckBox is a type of two state button either unchecked or checked in Android. For example, it can be used to know the hobby of the user, activate/deactivate the specific action etc.

Methods

- isChecked()-Returns true if it is checked otherwise false.
- setChecked(boolean status)- Changes the state of the CheckBox.

PROCEDURE

1. Create a new Android Project
2. Enter Application details
3. Create three checkbox instances and a button by declaring it inside xml file
4. In java file
    a. Create objects of Button and CheckBox class
    b. Set a click listener on the button object
    c. initiate checkboxes
    d. check current state of all checkboxes(true or false) by using isChecked() method .
    e. Display the text and amount of selected checkboxes and total amount using toast class.
5. Execute the application by clicking the menu Run -> Run

SOURCE CODE

MainActivity.java

/**********Write the java code**********/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here**********/**

RESULT

/*****************Write the result***************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 12 **Date :** D D / M M / Y Y

UNDERSTANDING RADIO BUTTON AND RADIO GROUP

AIM:

To familiarize attributes and methods of Radio Button and Radio Group class.

THEORY

In Android, RadioButton are mainly used together in a RadioGroup. In RadioGroup checking the one radio button out of several radio button added in it will automatically unchecked all the others. It means at one time we can checked only one radio button from a group of radio buttons which belong to same radio group. RadioButton is a two state button that can be checked or unchecked. If a radio button is unchecked then a user can check it by simply clicking on it. Once a RadioButton is checked by user it can't be unchecked by simply pressing on the same button. It will automatically unchecked when you press any other RadioButton within same RadioGroup.

PROCEDURE

1. Create a new Android Project
2. Enter Application details
3. Create two radiobutton inside a RadioGroup instance and a button by declaring it inside xml file
4. In java file
    a. Create objects of Button , RadioButton and RadioGroup class
    b. Set a click listener on the button object
    c. initiate declared objects
    d. check current state of all radiobuttons(true or false) by using isChecked() method .
    e. Display the text of selected radiobutton.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 13 **Date :** D D / M M / Y Y

EXPLICIT INTENT-I

AIM:

To familiarize how to navigate through activities using Intent.

THEORY

Android application components can connect to other Android applications. This connection is based on a task description represented by an Intent object. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. Explicit intents explicitly define the component which should be called by the Android system, by using the Java class as identifier.The following shows how to create an explicit intent and send it to the Android system to start an activity.

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
startActivity(i);
```

PROCEDURE

1. Start Android Studio and create a new Android Studio project.
   - Call application "Two Activities" and change the company domain to "android.example.com." Choose the same Minimum SDK that used in the previous projects.
   - Choose **Empty Activity** for the project template. Click **Next.**
   - Accept the default activity name (MainActivity). Click **Finish**.
2. Define the layout for the main activity
   - Open res/layout/activity_main.xml. In the Layout Editor.

- Click the Design tab at the bottom of the screen and change the TextView text as First Activity."
- Add a Button to the layout in any position.
- Switch to the XML Editor (click the Text tab) and modify these attributes in the Button:
  - `android:id="@+id/button_main"`
  - `android:text="Call Second Activity"`
  - android:onClick="launchSecondActivity"

3. Create the second activity
   - Click the **app** folder for your project and choose **File > New > Activity > Empty Activity**.
   - Name the new activity "SecondActivity." Check **Generate Layout File** and layout name will be filled in as activity_second.
   - Click **Finish**.

4. Modify the Android manifest.
   - Open `manifests/AndroidManifest.xml`.
   - Find the `<activity>` element that Android Studio created for the second activity. <activity android:name=".SecondActivity"></activity>
   - Add these attributes to the `<activity>` element: `android:label= "Second Activity"` & android:parentActivityName= ".MainActivity".
5. Define the layout for the second activity
   - Open res/layout/activity_second.xml and change the root view group to RelativeLayout
   - Add a TextView and set attribute text as “Second Activity”.
6. Add an intent to the main activity
   - Open the Java file for MainActivity
   - Create a new intent in the launchSecondActivity() method.

- Send data from the main activity to the second activity using putExtra() method.
- Call the startActivity() method with the new intent as the argument.

7. Modify the second activity to get the extras and display the message
    - Open java/com.example.android.twoactivities/SecondActivity.
    - In the onCreate() method, get the intent that activated this activity: using getIntent();
    - Get the string containing the message from the intent extras.
    - Toast the message.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/*****************Write the result***************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 14 **Date :** D D / M M / Y Y

EXPLICIT INTENT-RADIO BUTTON

AIM:

To understand passing of selected radiobutton and EditText value to another activity using Intent.

THEORY

Explicit intents explicitly define the component which should be called by the Android system, by using the Java class as identifier. The following code is to create an explicit intent and send it to the Android system to start an activity.

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
startActivity(i);
```

Get the intent that activated this activity:

```
Intent intent = getIntent();
String message = intent.getStringExtra("Value1");
```

OR

```
Bundle extras = getIntent().getExtras();
String message = extras.getString("Value1");
```

PROCEDURE

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

- Call application "ExplictII" and change the company domain to "android.example.com." Choose the same Minimum SDK that used in the previous projects.
- Choose **Empty Activity** for the project template. Click **Next.**
- Accept the default activity name (MainActivity). Click **Finish**.

2.Define the layout for the main activity

- Open res/layout/activity_main.xml. In the Layout Editor.
- Add a Button,TextView(text:enter your name),EditText and a RadioGroup(Enter your branch) with three radio buttons(text:MECH,CT &EC) to the layout in any position.
- Switch to the XML Editor (click the Text tab) and modify these attributes in the Button:
  - android:text="Submit"
  - android:onClick="launchSecondActivity"

3.Create the second activity

- Click the **app** folder for your project and choose **File > New > Activity > Empty Activity**.
- Name the new activity "SecondActivity." Check **Generate Layout File** and layout name will be filled in as activity_second.
- Click **Finish**.

4.Modify the Android manifest.

- Open manifests/AndroidManifest.xml.
- Find the <activity> element that Android Studio created for the second activity. <activity android:name=".SecondActivity"></activity>
- Add these attributes to the <activity> element: android:label= "Second Activity" & android:parentActivityName= ".MainActivity".

5.Define the layout for the second activity

Open res/layout/activity_second.xml and change the root view group to RelativeLayout

- Add two TextViews.

8. Add an intent to the main activity

- Open the Java file for MainActivity
- Create a new intent in the launchSecondActivity() method.
- Send data from the main activity to the second activity using putExtra() method.
- Call the startActivity() method with the new intent as the argument.

6.Modify the second activity to get the extras and display the message

- Open java/com.example.android.twoactivities/SecondActivity.
- In the onCreate() method, get the intent that activated this activity: using getIntent();
- Get the string containing the message from the intent extras using getStringExtra().
- Set the message to TextView.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

SAMPLE OUTPUT

/****Draw the obtained output here*************/**

RESULT

/*****************Write the result***************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 15 **Date :** D D / M M / Y Y

EXPLICIT INTENT- CHECKBOX

AIM:

To understand passing data of selected Checkbox and EditText value to another activity using Intent.

THEORY

Explicit intents explicitly define the component which should be called by the Android system, by using the Java class as identifier. The following code is to create an explicit intent and send it to the Android system to start an activity.

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
startActivity(i);
```

Get the intent that activated this activity:

```
Intent intent = getIntent();
String message = intent.getStringExtra("Value1");
                    OR
Bundle extras = getIntent().getExtras();
String message = extras.getString("Value1");
```

The selection of checkbox can be check with the method isChecked() and text can be retrieved by getText().

PROCEDURE

- Start Android Studio and create a new Android Studio project.
- Call application "ExplictIII" and change the company domain to "android.example.com." Choose the same Minimum SDK that used in the previous projects.

- Choose **Empty Activity** for the project template. Click **Next.**
- Accept the default activity name (MainActivity). Click **Finish**.

1. Define the layout for the main activity

   - Open res/layout/activity_main.xml. In the Layout Editor.
   - Add a Button,two TextViews,EditText and three CheckBoxes to the layout in any position.
   - Switch to the XML Editor (click the Text tab) and modify these attributes in the Button:
     - android:text="Submit"
     - android:onClick="launchSecondActivity"

2. Create the second activity

   - Click the **app** folder for your project and choose **File > New > Activity > Empty Activity**.
   - Name the new activity "SecondActivity." Check **Generate Layout File** and layout name will be filled in as activity_second.
   - Click **Finish**.

3. Modify the Android manifest.

   - Open manifests/AndroidManifest.xml.
   - Find the <activity> element that Android Studio created for the second activity. <activity android:name=".SecondActivity"></activity>
   - Add these attributes to the <activity> element: android:label= "Second Activity" & android:parentActivityName= ".MainActivity".

4. Define the layout for the second activity
   - Open res/layout/activity_second.xml and change the root view group to RelativeLayout
   - Add two TextViews.
5. Add an intent to the main activity
   - Open the Java file for MainActivity
   - Retreive the value from selected checkboxes using isChecked() and getText().
   - Create a new intent in the launchSecondActivity() method.
   - Send data from the main activity to the second activity using putExtra() method.
   - Call the startActivity() method with the new intent as the argument.
6. Modify the second activity to get the extras and display the message
   - Open java/com.example.android.twoactivities/SecondActivity.
   - In the onCreate() method, get the intent that activated this activity: using getIntent();

- Get the string containing the message from the intent extras using getStringExtra().
- Set the messages to TextViews using setText() method.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 16 **Date :** D D / M M / Y Y

RESET BUTTON

AIM

To understand retrieve value from EditText and set value to the TextView on same activity.

THEORY

EditText is a subclass of TextView with text editing operations. We often use **EditText** in our applications in order to provide an input or text field, especially in forms.**getText()** method to get the text entered in the EditText views. But getText()method returns an Editable instance and therefore we have **typecasted** it to convert it into String for further use. This can be done by using the toString() method.

In Android, TextView displays text to the user and optionally allows them to edit it programmatically. **TextView code in JAVA:**

```
TextView textView = (TextView) findViewById(R.id.textView);

textView.setText("AbhiAndroid"); //set text for text view
```

PROCEDURE

Start Android Studio and create a new Android Studio project.

- Call application "ExplictII" and change the company domain to "android.example.com." Choose the same Minimum SDK that used in the previous projects.
- Choose **Empty Activity** for the project template. Click **Next.**
- Accept the default activity name (MainActivity). Click **Finish**.

Define the layout for the main activity

- Open res/layout/activity_main.xml. In the Layout Editor.
- Add a Button,four TextViews,two EditText to the layout in any position.

In java file

- Create objects of Button ,EditText and TextView class
- Set a click listener on the submit button object
- Retrieve the values from two EditText.

- Invoke setText() method of EditText class to display it.
- Set a click listener on the reset button object
- Invoke setText() method of EditText class to reset it
- Execute the application by clicking the menu Run -> Run

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 17 **Date :** D D / M M / Y Y

MAKE A PHONE CALL

AIM:

To implement phone call in android studio

THEORY

Android provides Built-in applications for phone calls, in some occasions we may need to make a phone call through our application. This could easily be done by using implicit Intent with appropriate actions. Also, we can use PhoneStateListener and TelephonyManager classes, in order to monitor the changes in some telephony states on the device. ACTION_CALL action trigger built-in phone call functionality available in Android device. Following is simple syntax to create an intent with ACTION_CALL action

```
Intent phoneIntent = new Intent(Intent.ACTION_CALL);
```

ACTION_DIAL action can be used instead of ACTION_CALL, in that case we will have option to modify hardcoded phone number before making a call instead of making a direct call.To make a phone call at a given number 91-000-000-0000, we need to specify tel: as URI using setData() method as follows −

```
phoneIntent.setData(Uri.parse("tel:91-000-000-0000"));
```

To make a phone call, Android need **CALL_PHONE** permission .In Manifest.xml file add this

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

PROCEDURE

1.Open Android Studio and start create a new project with an empty activity called MainActivity

2.Add a TextView,EditText and button on layout

3. To make call requests, add call permissions in AndroidManifest.xml.

4.In Java code, implement intent with ACTION_CALL.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No : 18 **Date :** D D / M M / Y Y

STUDENT DATABASE

AIM:

To learn how to insert, up, delete and retrieve records in SQLite Database in Android

THEORY

SQLite is used as a database for android application development. SQLite is a Structure query base database, open source, light weight, no network access and standalone database. SQLite is RDBMS (Relational Database Management System), is written in C programming language.SQLite is embedded within the Android operating System, no need of anything external on Android to use SQLite.To manipulate data (insert, up , delete) in SQLite database – we'll use SQL (Structured Query Language)

PROCEDURE

1.Create a new project in android studio
2.Create layouts.
3. Now create a new Java class called DatabaseHelper.java**.**
4. In DataBaseHelper.java

- Define name of Database, table and columns
- Define Constructor
- Define create table query inside onCreate() method.
- Define onUpgrade() method.
- To insert data,
- open database
- Create an instance of a class ContentValues and use put method to store values in the object
- Insert data in the table using insert method on the SQLiteDatabase instance
- To delete data
- Open database for deleting data in the tables using getWritableDatabase()
- Delete data from table using delete method on the SQLiteDatabase instance
- To up data
- Open database for updating data in the tables using getWritableDatabase()
- Create an instance of a class ContentValues and use put method to store values in the object
- Up data in the table using up method on the SQLiteDatabase instance

- To read data from database,
- Open database for updating data in the tables using getWritableDatabase()
- Write the query for reading values in the method rawQuery().

5. In Main Activity.java

- Create an object of Class DataBaseHelper
- Create objects of UI Components.
- Call methods to insert ,delete,up and read data on respective button click.

SOURCE CODE

MainActivity.java

/**********Write the java code*****************/

Activity_main.xml

/*********Write xml code to display “hello”*******/

OUTPUT

/****Draw the obtained output here*************/**

RESULT

/******************Write the result****************/

	Signature of Lab in Charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exercises

1. Create an app to display the following

```
    *
   **
  ***
 ****
*****
```

2. Implement different layouts in Android Studio.
3. Create an activity to accept details of a student such as Roll No and Name. Toast the details on same activity.
4. Create an activity to accept two numbers. Toast the sum on same activity.
5. Create an activity to accept details of a student such as Roll No and Name. Display the details on another activity.
6. Create an activity to input details of a student such as Roll No, Name and branch should be selected from (Mech, EC & Computer). Toast the details on same activity.
7. Create an activity to choose your favourite books. Toast the details on another activity.
8. Create a simple List View
9. Create a simple List View with image and text
10. Integrate a website inside the application using Webview
11. Create an employee database and prepare pay bill.

****