# 1) <u>Introduction   to J2ME</u>

J2ME (Java 2 Micro Edition) is an advanced technology in Java, developed with the help of Java Community Process Program. J2ME is a reduced version of the Java API and Java Virtual Machine that is designed to operate within the limited resources available in the embedded computers and microcomputers.

J2ME is targeted to developers of intelligent wireless devices and small computing devices that need to incorporate cross-platform functionality in their products. A key benefit of using J2ME is compatibility with all Java-enabled devices. Motorola, Nokia, Panasonic all have Java-enabled devices. A J2ME application is a balance between local and server-side processing.

The Java Community Process Program used a two approaches to addressing the needs of small computing devices.

Configurations: It is the Java run-time environment and core classes that operate on each device. A configuration defines the Java Virtual Machine for a particular small computing device. There are two configurations.

CLDC for handheld devices: The CLDC (Connected Limited Device Configuration) is designed for 16-bit or 32-bit small computing devices with limited memory. These devices usually have between 160KB and 512KB of available memory. Usually these are powered by battery. They use small bandwidth network wireless connection. These devices use a stripped-down version of the JVM the KJava Virtual Machine (KVM). These devices include pagers, personal digital assistants, Cell phones, dedicated terminals, and handheld consumer device.

CDC for plug-in devices: CDC (Connected Device Configuration) devices use a 32-bitarchitecture,have at least 2 MB of memory available, and implement a complete functional JVM. CDC devices include digital set-top boxes, home appliances, navigation systems, point-of-sale terminals, and smart phones.

Profiles: It is defined for categories of small computing devices. A profile consists of classes that enable developers to implement features found on a related group of small computing devices. List of J2ME
Profiles:
1. Profiles Used with CLDC:
2. Mobile Information Device Profile (MIDP)
3. PDA Profile (PDAP)
Profiles Used with CDC:

Foundation Profile
        o Game Profile
        o Personal Profile
        o Personal Basis Profile
        o RMI Profile.

### J2ME Architecture

There are 5 layers in J2ME Architecture.
Those are:
MIDP (Top Most Layer): Which contains Java APIs for user network connections, persistence storage and the user interface? It also has access to CLDC libraries and MIDP libraries.
J2ME API's (Profiles): Which consists of the minimum set of application programming interfaces for the small computing device.
Configurations: Which handles interactions between the profile and the JVM?
JVM Operating System (Bottom Layer).
1,2 and 3 are software Layers.
3) Midlets.
A MIDlet is a J2ME application which operates on an MIDP. A MIDlet is defined with at least a single class that is derived from the javax.microedition.midlet.MIDlet abstract class.
Common programming is grouping related MIDlets into a MIDlet suite, which is contained within the same package and implemented simultaneously. All MIDlets within a MIDlet suite are considered a group and must be installed and uninstalled as a group.

MIDlets from the same MIDlet suite run the same class. Benefit of the relationship among MIDlet suite members is that they share the same data.

A MIDlet is an event-based application. All routines executed in the MIDlet are invoked in response to an event reported to the MIDlet by the application manager. The initial event that occurs is when the MIDlet is started and the application manager invokes the startApp() method.

The startApp() method in a typical MIDlet contains a statement that displays a screen of data and prompts the user to enter a selection from among one or more options. The nature and number of options is, MIDlet and screen dependent.

A Command object is used to present a user with a selection of options to choose from when a screen is displayed. Each screen must have a CommandListener.

A Command Listener monitors user events with a screen and causes the appropriate code to execute based on the current event.

A MIDlet is a class that extends the MIDlet class and is the interface between application statements and the run-time environment, which is controlled by the application manager. A MIDlet class must contain three abstract methods that are called by the application manager to manage the life cycle of the MIDlet. These abstract methods are.

startApp(): called by the application manager when the MIDlet is started and contains statements that are executed each time the application begins execution. Public and have no return value nor parameter list.

pauseApp(): called before the application manager temporarily stops the MIDlet. The application manager restarts the MIDlet by recalling the startApp() method. Public and have no return value nor parameter list.

destroyApp(): called prior to the termination of the MIDlet by the application manager. Public method without a return value. It has a boolean parameter that is set to true if the termination of the MIDlet is unconditional, and false if the MIDlet can throw a MIDletStateChangeException.

## **Installing and Working with J2ME Wireless Toolkit**

1) If the Java Development Kit (JDK) is not there or only having the Java Runtime Environment (JRE) installed, install the latest JDK from   http://java.sun.com/javase/downloads/index.jsp. Current stable release of Java is JDK 6 Update 7 but check the web page in case there are newer non-beta releases available.

2) Next, download the **Java Wireless Toolkit** (formerly called J2ME Wireless Toolkit) from: http://java.sun.com/products/sjwtoolkit/download.html.

3) Run the installer (for example, for Windows it is: sun_java_wireless_toolkit- 2_5_2-windows.exe). The installer checks whether a compatible Java environment has been preinstalled.

If not, it is necessary to uninstall old versions of Java and perform Step 1 again.

Once after successful installation of Java and the tool kit compile this program and run the following program in the toolkit.

Steps to run this program in toolkit:

1. Start -> All Programs -> Sun Java Tool Kit -> Wireless Tool Kit

2. Click New Project – Enter Project Name -> Enter Class Name -> Click on Create Project. 3. Choose appropriate API Selection and Configurations.

4. Place Java Source file in WTK2.1 / WTK2.2\apps\projectname\src
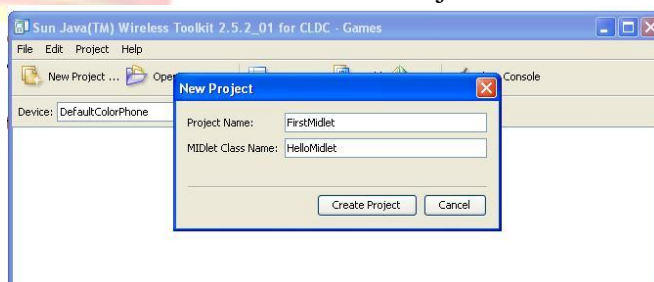
5. Build the Project.

6. Run the Project.

**P1**:- **Create a program to display HelloWorld message**
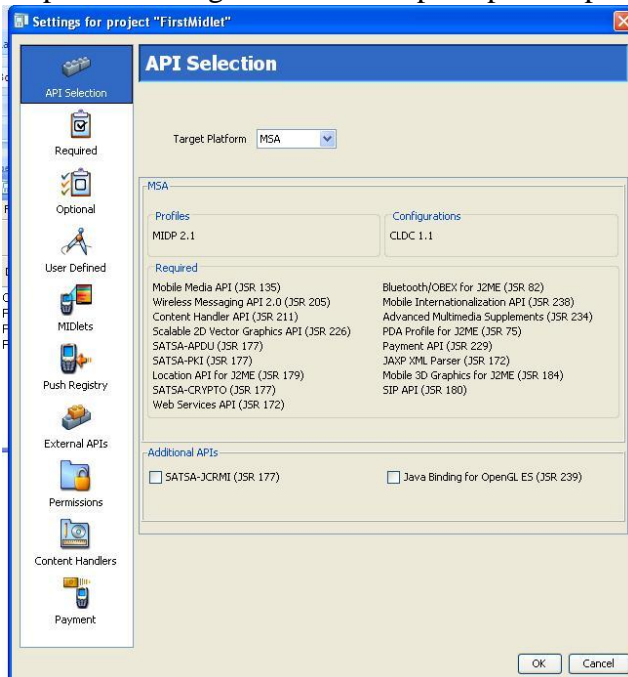
**Algorithm:-**

Step-1:-Start ->AllPrograms->Sun Java Tool Kit->Wireless Tool Kit
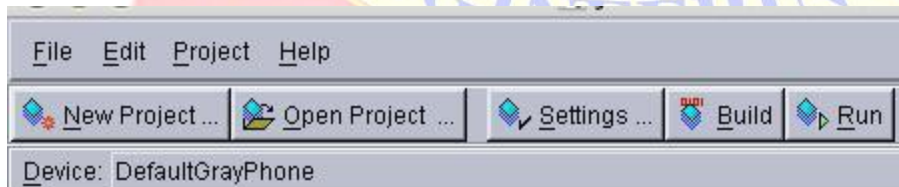


Step-2:-Click New Project –Enter project Name as FirstMidlet -> Enter ClassName as HelloMidlet->click on Create Project

Step 3:  A setting window will open up. Accept the defaults by clicking ok in that window.



1.Import the packages needed to run the MIDlet

2.declare the variables needed in the MIDlet 3.Start the

life cycle of MIDlet

4.Write the methods required

startApp(),pauseApp(),destroyApp(); 5. write the

message into the Form 6.Display the message in

the emulator

**1. Write a J2ME program to show how to change the font size and color.**

**Aim:**

To Write a J2ME program for showing the change font size and color.

**Algorithm**

1. Import the packages needed to run the MIDlet and implements the interface needed

2.declare the variables needed in the MIDlet

3. Start the life cycle of MIDlet

4. Write the methods required

startApp(),pauseApp(),destroyApp();

 5. Give the color items required

6. Start the life cycle

7. Add the menu to the emulator

8. Run the life cycle of J2ME

**Output:** the output can observe on mobile screen with different color and font size of text.
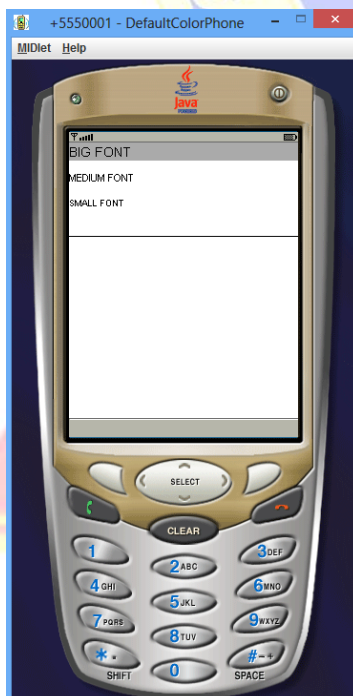
**Conclusion:** Student get the Knowledge about Mobile devices which is based on J2ME and after this experiment can able to handle with small applications related J2ME like changing the color and font size of screen and words respectively. This experiment maps with CO1, CO2, CO3, CO4, CO6 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

**Viva Voce Questions**

1.What is J2ME?
2.What is the J2ME platform?
3.J2ME defines two configurations - CLDC and CDC. Explain them
4.What are profiles in J2ME?
5.What is Java Community Process (JCP)?
6. Waht is Canvas in J2ME?
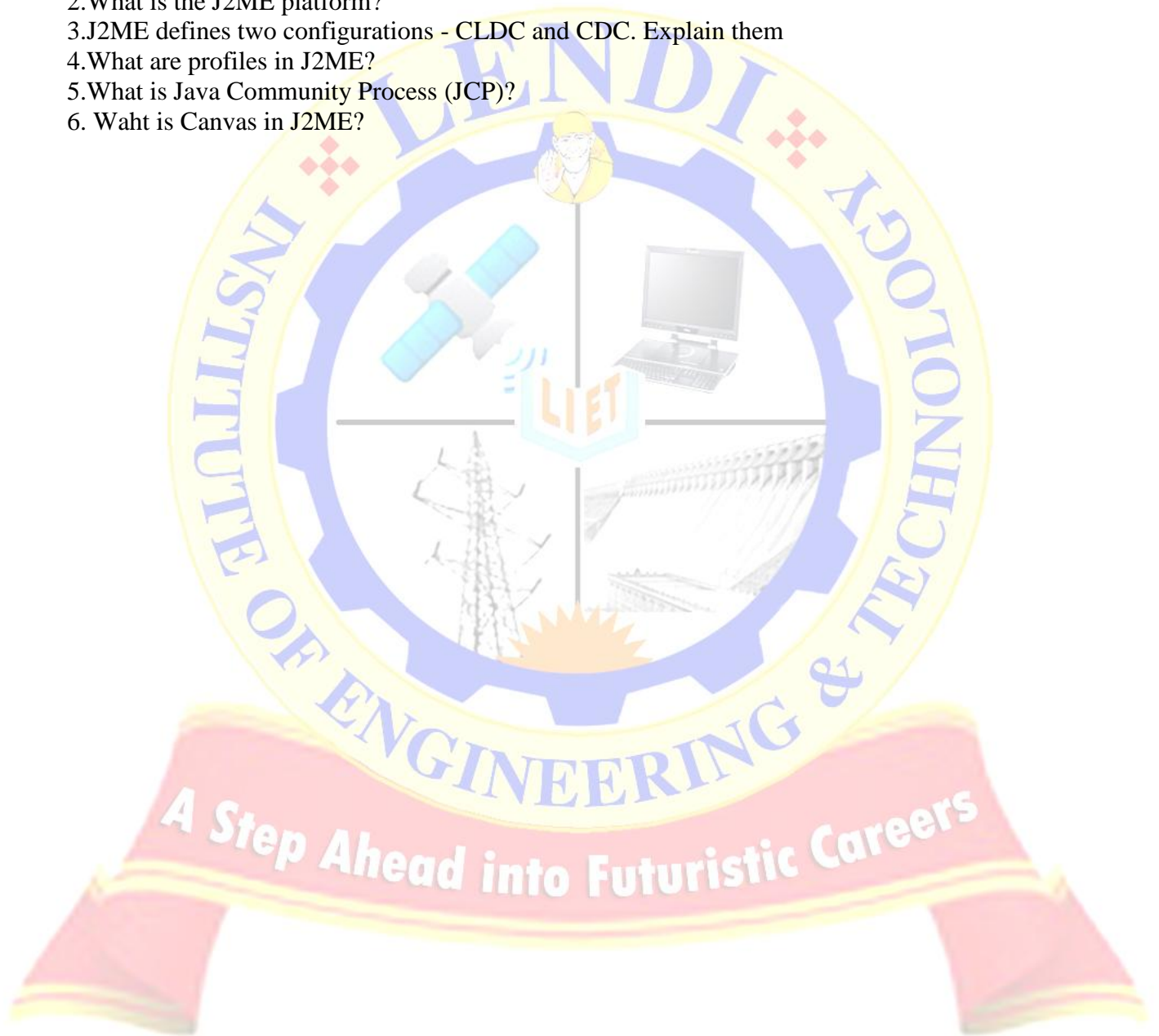
**2. Create a program which creates to following kind of menu.**

* cut
* Copy
* Paste
* delete
* select all
* unselect all

**<u>Aim</u>**

To create a J2ME Program this creates menu bar as per the specifications.

**<u>Algorithm</u>**

1.Import the packages needed to run the MIDlet and implements the interface needed 2.declare

the variables needed in the MIDlet

3.Start the life cycle of MIDlet

 4.Write the methods required

startApp(),pauseApp(),destroyApp();

5. give the menu items required

radioButtons = new ChoiceGroup( "Select Your Color", Choice.EXCLUSIVE);

6.Add the menu to the emulator

7.radioButtons.append("copy", null);

8. Run the life cycle of J2ME

**Output:** As per the given input in the question



**Conclusion:** Thus the student could able to understand small memory based devices and can able to implement creation of menu on screen in J2ME based devices. This experiment maps with CO1, CO2, CO3, CO4, CO5 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

**Viva Voce Questions**

1.What is LCDUI

2.What is MIDlet

3.What is MIDlet suite?

4. What is MIDP

5. What is J2ME WTK ?

6. What are the different types of possibilities create a menu ?

### 3. Create a program which creates a select menu for Event handling

Create a menu which has the following options:

* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

**Aim :** To create a J2ME program that creates a select menu for event handling as per the given specifications.

### Algorithm

1. Import the packages needed to run the MIDlet and implements the interface needed

2.declare the variables needed in the MIDlet

3. Start the life cycle of MIDlet

4. Write the methods required

startApp(),pauseApp(),destroyApp();

 5. Give the color items required

6. Start the life cycle

7. Add menu to the emulator

8. Perform the actions

**Output:**



**Conclusion**: Thus the student can get knowledge about creation of menu using event handling on mobile screen and can select the events for the menu. This experiment maps with CO1, CO2, CO3, CO4, CO6 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

**Viva Voce Questions?**

1) Describe the menubar in J2ME?
2) Describe Menu component hierarchy?

**4. Create an MIDP application which draws a bar graph to display. Data values can be given at int[]array?**

**Aim:** To create a MIDP application that draws a bar graph to display using int[] array.

**Algorithm:-**

1. Import the packages needed to run the MIDlet and implements the interface needed

2.declare the variables needed in the MIDlet

3. Start the life cycle of MIDlet 4.Write

the methods required

startApp(),pauseApp(),destroyApp(); 5.

give the menu items required

6. Start the life cycle

7. Add the menu  to the emulator
Step-1:-Start ->AllPrograms->Sun Java Tool Kit->Wireless Tool Kit
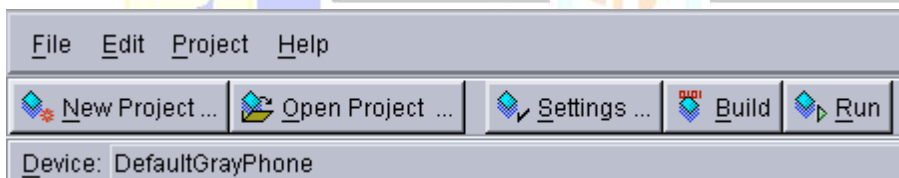


Step-2:-Click New Project –Enter project Name as FirstMidlet -> Enter ClassName as HelloMidlet->click on Create Project
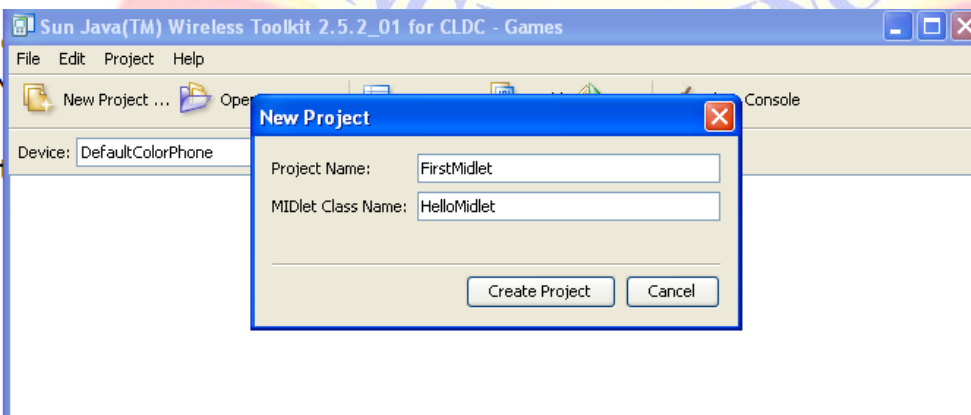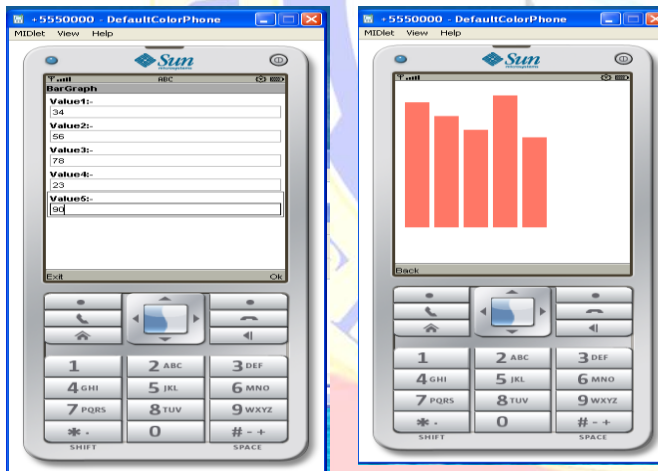
Step 3: click settings button and select midlet- >add give midlet name , class name
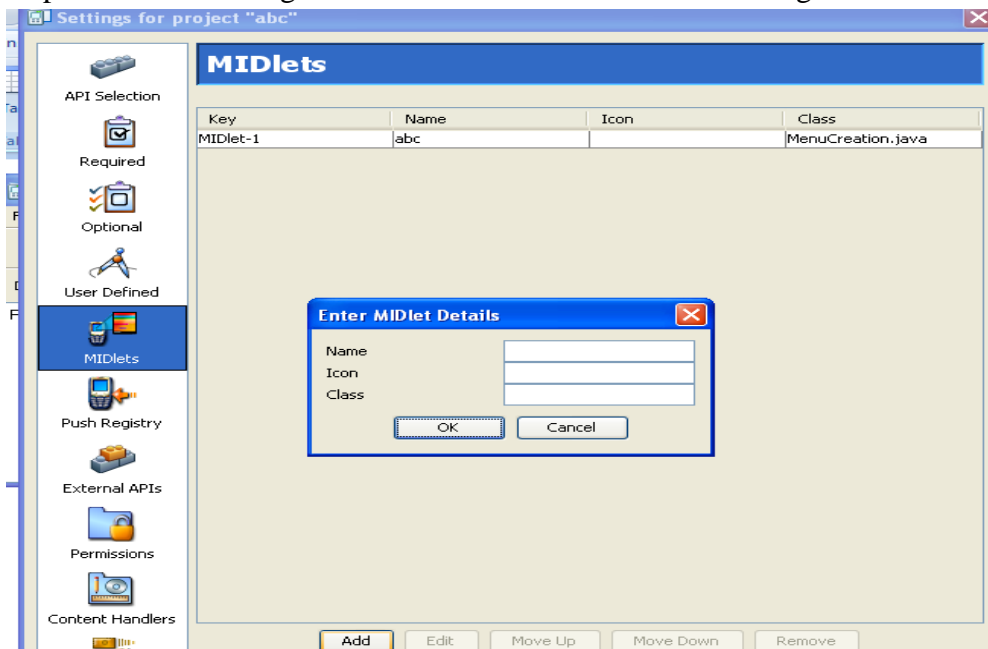


Step 4: click BUILD button
Step 5: if build successfully completed then click RUN button

**Output:**



**Conclusion:** Thus the student can get the knowledge to creating the bar graph using canvas class in J2ME. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

**Viva Voce Questions**

1) What is canvas in public abstract class Canvas extends Displayable

2) What is Displayable in public abstract class Displayable extends Object?

**5. Create an MIDP application which examines a phone number, that a user entered in a given format.**
**\*Area code should be one of the following: 040, 041, 050, and 0400,044**
There should 6-8 numbers in telephone number (+area code).

**Aim** : To create an MIDP application that examines a phone number as per the given format.

## Algorithm

1. Import the packages needed to run the MIDlet and implements the interface needed

2.declare the variables needed in the MIDlet

3. Start the life cycle of MIDlet

4. Write the methods required

startApp(),pauseApp(),destroyApp();

 5. Give the color items required

6. Start the life cycle

7. Add menu to the emulator

8. Perform the actions

## Output:
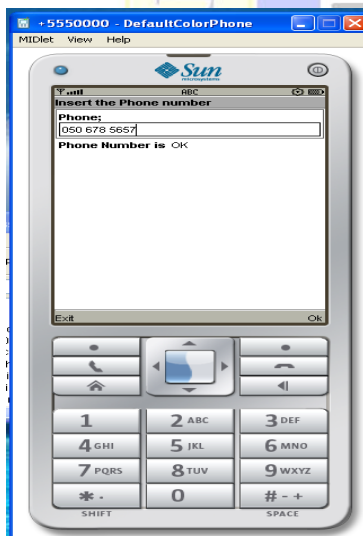


**Conclusion:** Thus the student can get the knowledge of checking input  using array in J2ME and can test various inputs for checking as per given question. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

## Viva Voce Question

   1) What is MIDlet?
   2) What is the Life cycle MIDlet?

**6. Write a sample program to show how to make a SOCKET Connection from J2ME phone. This J2ME sample program shows how to how to make a SOCKET Connection from a J2ME Phone. Many a times there is a need to connect backend HTTP server from the J2ME application. Show how to make a SOCKET connection from the phone to port 80.**

**Aim** To write a sample program that shows simple socket connection from J2ME phone

**Algorithm**

1. Import the packages needed to run the MIDlet 2.declare the

variables needed in the MIDlet 3.Start the life cycle of MIDlet

4. Write the methods required

startApp(),pauseApp(),destroyApp();

5. Create server side program by implementing the interfaces command listener,Runnable

6.declare the required variables needed.

7. Use start, run, synchronized void run() methods.

8. Create the socket connections in the run method.

9. create Client side program by implementing the interfaces command listener,Runnable

10. Declare the required variables needed.

11.  use start, run, synchronized void run() methods

12. Create the socket connections in the run method

13. Close the socket connections.

**Output:**



**Conclusion:** Thus the student can get the knowledge of handling the socket connection between the server and client for J2ME based mobiles. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

**Viva Voce Questions**

1. What is SOAP ?

2. What is SSL ?

3. What is Tomcat ?

4. What is UDDI ?

### 7. Login to HTTP Server from a J2ME Program

This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server. Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.

**Note**: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

Login to HTTP Server from a J2ME Program

### Input & Output
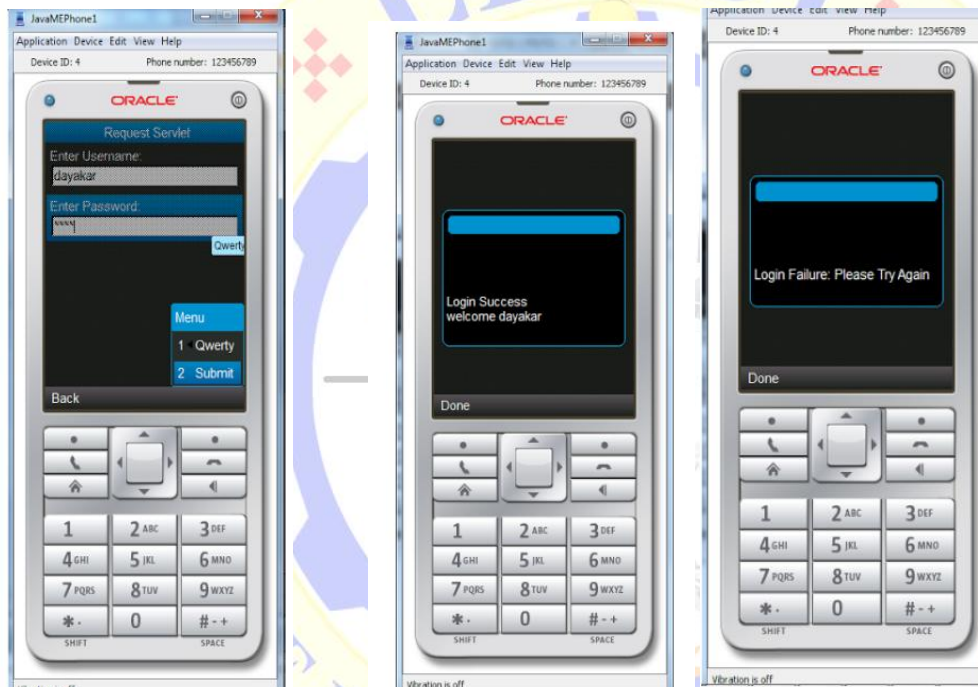


**Conclusion :** Thus the student get knowledge about the connection between client and server with login credentials inJ2ME. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, & PSOS: PSO1

### Viva Voce Questions

1. What is HTTPS?

2. What is cHTML ?

3. What is Personal Profile

# 8.ANDROID

Android is Google's oft-discussed mobile, wireless, computer, and communications platform. You can take advantage of the powerful Eclipse environment to build Android applications using the Android Eclipse plug-in.

System requirements.

## Eclipse Platform:
　　　Eclipse is the platform upon which the plug-in runs. Get the latest version of Eclipse Classic (V3.3.1 was used in this tutorial).
Android Developer Tools
The Android Developer Tools (the Eclipse plug-in) may be installed by following the instructions found at Installing the Android SDK.
Source code

**AndroidManifest.xml snippet** — This file is the application deployment descriptor for Android applications.

**IntentReceiver** — This demonstrates the implementation of an IntentReceiver, which is the class that processes intents as advertised by the IntentFilter tag in the AndroidManifest.xml file.

**SaySomething.java** — This implements an Android activity, the primary entry point to the sample application.

**Main.xml** — This contains the visual elements, or resources, for use by Android activities.

**R.java** — This file is automatically generated by Android Developer Tools and "connects" the visual resources to the Java source code.

**AndroidManifest.xml complete** — This lists a full AndroidManfest.xml file, along with a description of each of the important elements.

**MobileServiceCallContacts.java** — This contains the code necessary to display contacts as well as react to user input to subsequently perform a Google Maps address lookup.

## Android terminology :

Android application development under the Eclipse environment requires knowledge of the Eclipse environment and the Android platform. An understanding of the terms below is helpful in Android application development with the Eclipse plug-in.

### Open Handset Alliance
　　　This is the organization led by Google Inc., consisting of numerous public and private organizations.
Android
The flagship product of the Open Handset Alliance. This is an open source operating environment targeted for mobile devices.

### Emulator

A software tool representative of another system — This is often an environment that runs on a personal computer (IBM®, Mac, Linux®) that emulates another environment, such as a mobile-computing device.

### Linux

An open source operating system kernel at the heart of many computing platforms, including servers, desktop computers, networking appliances, and mobile-computing devices. Android runs on top of a Linux kernel.

### Dalvik Virtual Machine

The Dalvik VM is an operating environment found in the Android stack, which interprets application code at runtime. The Dalvik VM is similar to a compliant Java VM, but the two are not compatible.

### Android basics and required tools

Android is an open source operating system targeted for mobile platforms. At the time of this writing, it is a software-only platform with no publicly available hardware devices.
The Android platform is best described as a stack because it is a collection of components, including:
• Linux kernel-based operating system
• Java programming environment
• Tool chain, including compiler, resource compiler, debugger, and emulator
• Dalvik VM for running applications
Now that we've briefly introduced the Android platform architecture, let's look at some important characteristics of the platform from a market perspective.

### Architecture of Android:
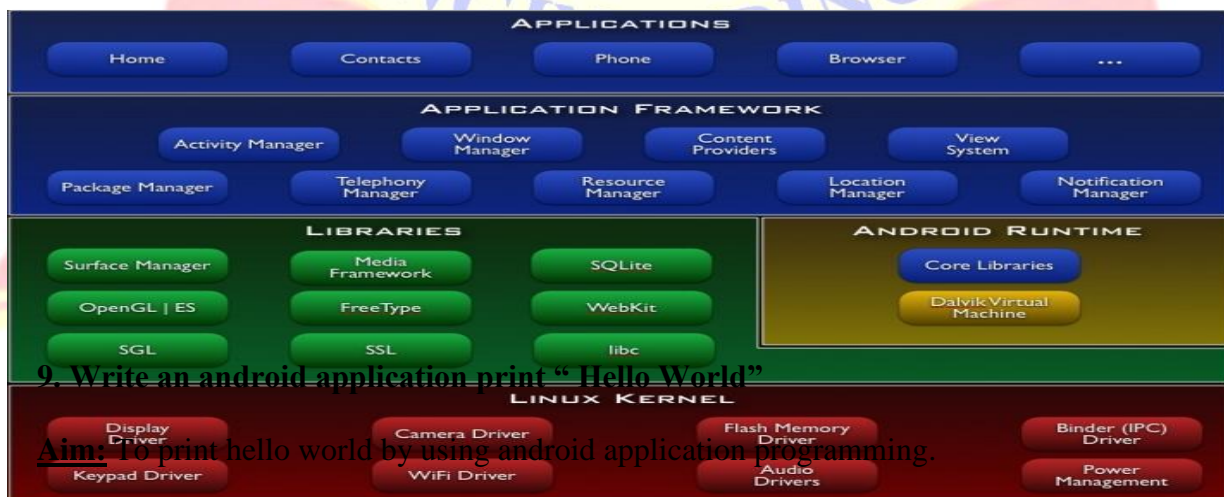


**9. Write an android application print " Hello World"**

**Aim:** To print hello world by using android application programming.

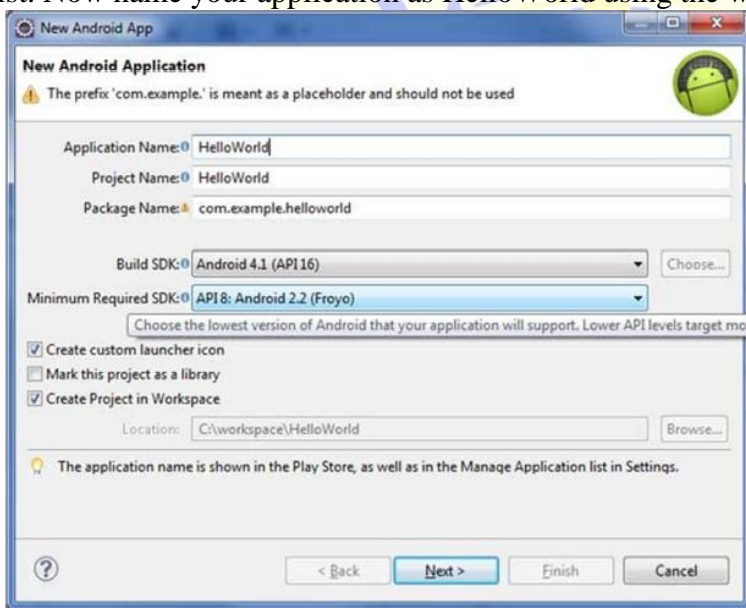**AIM: To Write an Android Application print " Hello World"**

Using Android Application print " Hello World"
Android SDK, you have to make sure that you have setup your Android development environment properly as explained in Android - Environment Setup. I also assume that you have a little bit working knowledge with Eclipse IDE. So let us proceed to write a simple Android Application which will print "Hello World!". Create Android Application
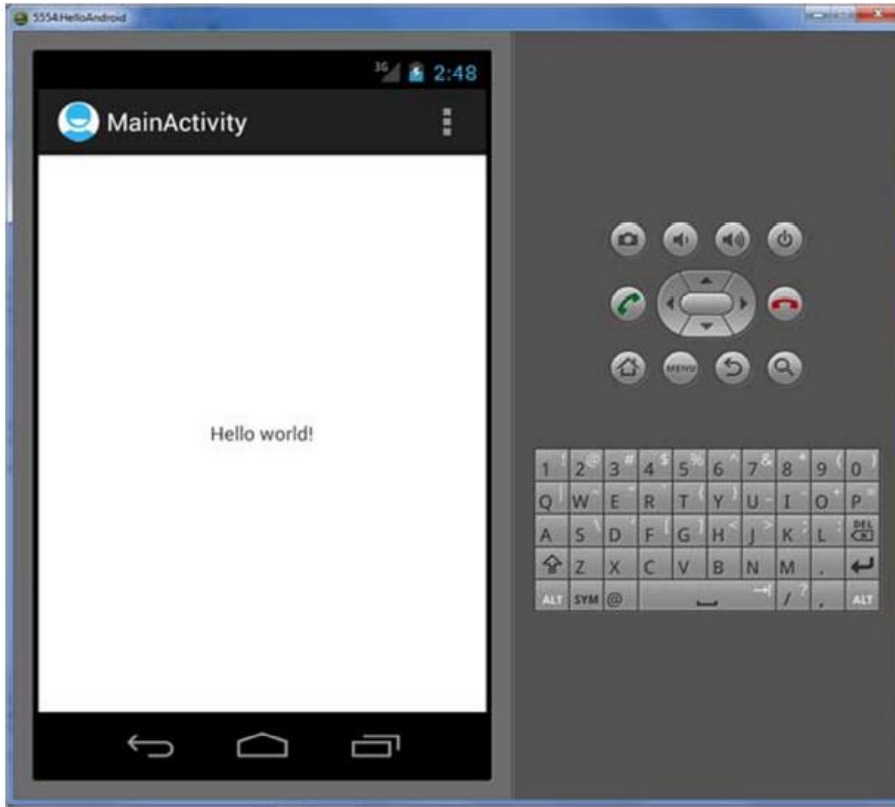
**ALGORITHM**
The first step is to create a simple Android Application using Eclipse IDE. Follow the option

File -> New -> Project and finally select Android New Application wizard from the wizard list. Now name your application as HelloWorld using the wizard window as follows:

## Conclusion:

Thus the student can get knowledge about simple android program and its execution lifecycle of android. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

## Viva Voce Questions

1) What is Android?
2) What Is the Google Android SDK?
3) What is the Android Architecture?

### 10. Write an android application using eclipse to print Hello World.

**Aim** To write a simple android application program for printing a hello world using Eclipse.

### Procedure:
1. Open Eclipse
2. Open the Emulator
[if you don't have created an AVD then create one using below few steps otherwise you can skip tillStep3]
*Go to Window menu and click on AVD Manager, which will open Android Virtual Device Manager in a new window.
*Click on New button
*A new window will be opened and will ask for Name.
*Type a name there.[Ex: MyDevice]
*Then choose target android version from dropdown Target. [Ex: Android 2.2 -API Level 8]
*Now click on Create AVD
*OK, your AVD is now ready to run which you can see in the AVD list on the AVD Manager window.
*Now, choose the device you just have created and then click on Start button.
3. Go to File –>New
4. Click on Project.
5. Select Android Project from Android directory/folder.
6. Click Next button.
7. Type your project name. [Ex: HelloWorld]
8. Click Next.
9. Select a build target. [Ex: Android 2.2]
10. Click Next.
11. Type a valid Package Name. [Ex: com.demo.mydemo]
12. Make sure that Create Activity checkbox is checked.
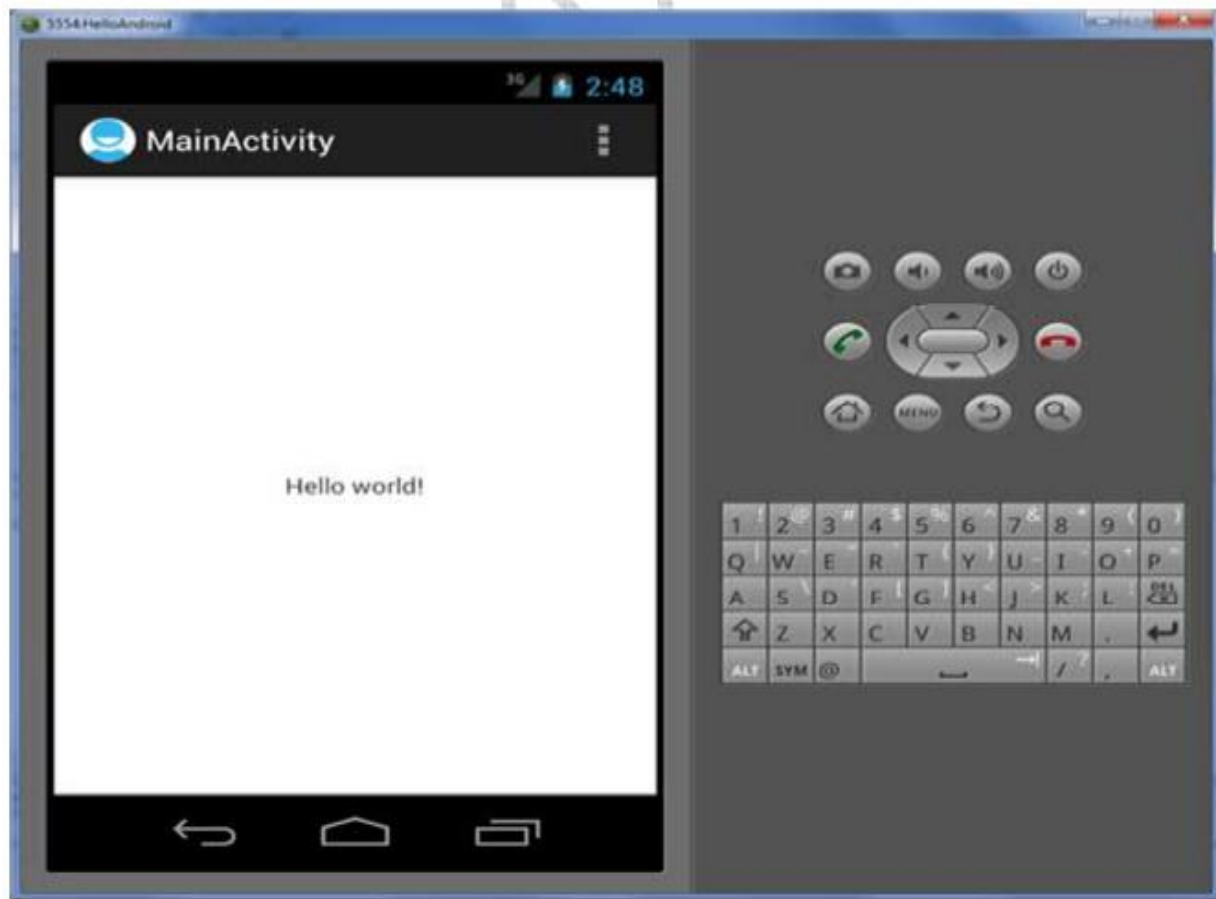13. Click Finish.
14. Now the project can be seen, listed on the Navigator.
15. It contains some pre-generated files and folders.
16. Open the Activity file located under src folder .[Ex: src --> com --> demo --> mydemo -->]
17. Delete the entire contents from it.

**Output:**



**Conclusion**

Thus the student can get the printing something on screen using Android programming and also its life cycle. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

**Viva Voce Questions**

1) What items are important in every Android project?
2) What is the importance of XMLbased layouts?
3) What are containers?
4) What is adb?
5) How are escape characters used as attribute?

**11. Write an Android application program that accepts a name from the user and displays the hello name to the user in response as output using Eclipse**

**Aim** To write a an android application program that accepts a name from the user and displays the

Hello name to the user in response as out using eclipse.

## **Procedure**

1. Open Eclipse
2. Open the Emulator [if you don't have created an AVD then create one using below few steps otherwise you can skip till Step3]
   *Go to Window menu and click on AVD Manager, which will open Android Virtual Device
    Manager in a new window.
   *A new window will be opened and will ask for Name.
   *Type a name there.[Ex: My Device]
   *Then choose target android version from dropdown Target. [Ex: Android 2.2 -API Level 8]
   *Now click on Create AVD
   *OK, your AVD is now ready to run which you can see in the AVD list on the AVD Manager
  Window.
   *Now, choose the device you just have created and then click on Start button.
3. Go to File –>New
4. Click on Project.
5. Select Android Project from Android directory/folder.
6. Click Next button.
7. Type your project name. [Ex: HelloWorld]
8. Click Next.
9. Select a build target. [Ex: Android 2.2]
10. Click Next.
11. Type a valid Package Name. [Ex: com.demo.mydemo]
12. Make sure that Create Activity checkbox is checked.
13. Click Finish.
14. Now the project can be seen, listed on the Navigator.
15. It contains some pre-generated files and folders.
16. Open the Activity file located under src folder .[Ex: src --> com --> demo --> mydemo -->]
17. Delete the entire contents from it.

**Output**



## Conclusion

Thus the student can get knowledge of that accepts a name from the user and displays the hello name to the user in response as output using Eclipse in Android programming. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

## Viva voce question

1. What is adb?
2. What is the Google Android SDK?
3. What is Android Architecture?
4. How can we use escape characters as attributes?

### 12 Write Android application program that demonstrates the following

### a)Linear Layout b)Relative Layout c)Table Layout d)Grid View Layout

**Aim :** To write an android application program that demonstrates the different types layouts

### Procedure

A Layout that arranges its children in a single column or a single row. The direction of the row can be set by calling setOrientation (). You can also specify gravity, which specifies the alignment of all the child elements by calling setGravity () or specify that specific children grow to fill up any remaining space in the layout by setting the weight member of   Linear   Layout. Layout programs. The default orientation is horizontal.

### ALGORITHN:

### Linear Layout

Linear layout can be created in two ways 1. GUI  2. Manually

1. GUI
Create an AndroidApplicationProject and expand the Project,
Expand the RES you can find the Layout, Expand the Layout and you can find the "activity_main.xml" auto generated file at the creation of project

2. Select the Layout and Right-click choose new ☐ other ☐ Expand Android ☐ choose AndroidXML Layout File

3. Select Linear Layout it should have small finish and name it with your letter at starting of interested name but the name and click
4.   Add the components in either      vertically setVerticalOrientation or linear layout you can set or horizontally by setHorizontalOrientation
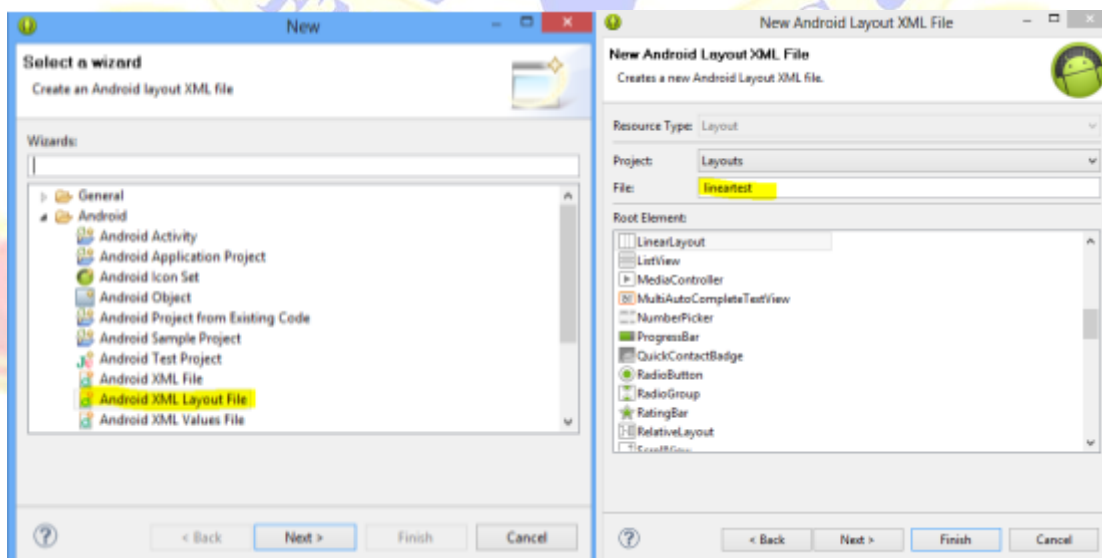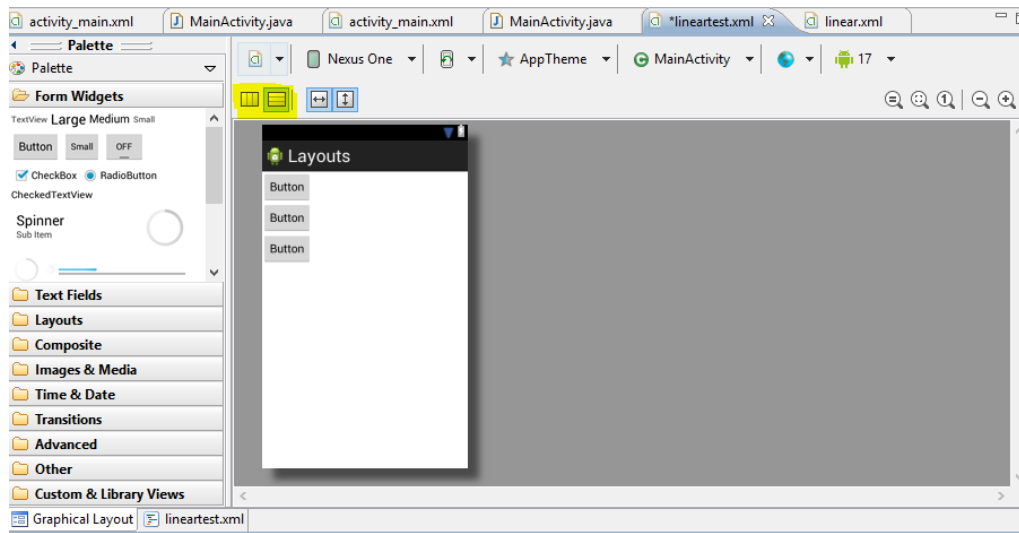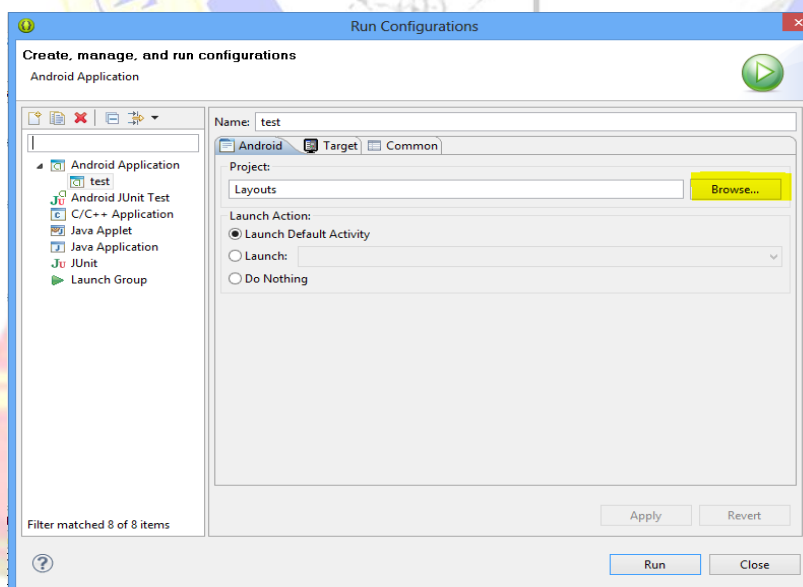
Step: 5

Select    SRC    expand    the    packed    and    open    "MainActivity.java"    and    change
setContentView(R.layout.linear); to your created layout name

Step :6

Run    the    application    by    right-clicking    "MainActivity.java"    and    select
"Runas"□ "RunCofigaration" browse your project Name and click run

**Input & Output**





**B) Relative Layout:**

In android , Relative Layout let you position your component base on the nearby component's position. It's the most flexible layout, that allow you to position your component to display in anywhere you want.

In relative layout, you can see "above, below, left and right" to arrange the component position, for example , display a button1 below button2 or display button3 on right of the button1.

We show you how to arrange position button, textview and editbox via "Relative Layout".

**ALGORITHM:**

Step1:
Create a new layout file Layouts->new->others->AndroidXMLLayoutFile
Note: You have use small letter for the starting letter of the file.

Step 2:
Update the "MainActivity.java" in the src folder that you have to change the layout name as seContentView(R.layout.relative)

Step 3:

Run the project by selecting project folder and right click choose option Runas->AndroidApplication.

Input & Output
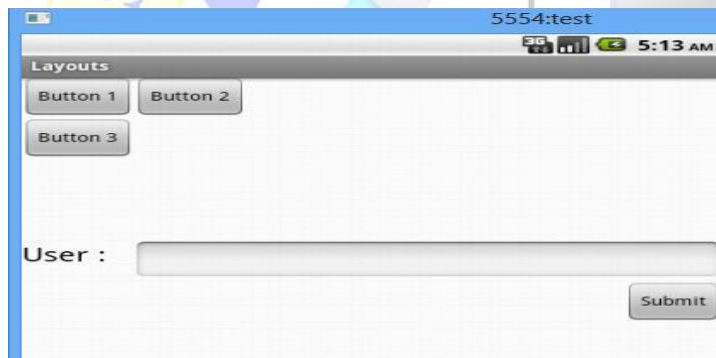


**C) Table Layout:**
In Android, TableLayout let you change components in rows and columns, just like the standard table layout in HTML, <tr> and <td>.

**ALGORITHM:**
Step1:
Create a new layout file Layouts->new->others->AndroidXMLLayoutFile
Note: You have use small letter for the starting letter of the file.

Step 2: Write the Content in xml file

Step 3:
Update the "MainActivity.java" folder in the src that you have to change the layout as setContentView(R.layout.table).

Step 4:
Run the project by selecting project folder and right click choose option
Runas->AndroidApplication

**Input & Output**
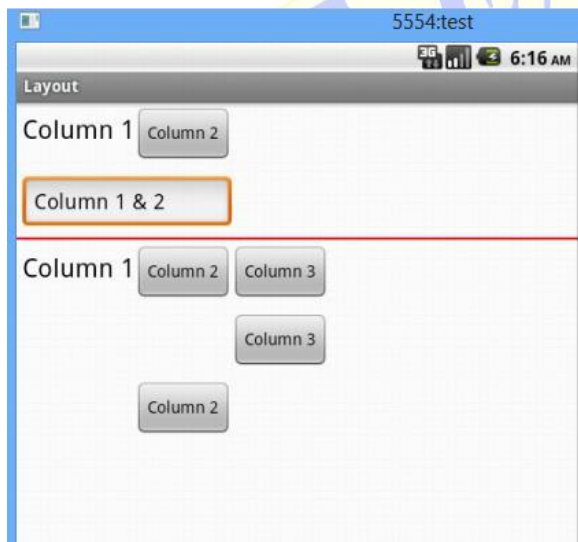


Grid Layout:

In Android, GridView let you arranges components in a two-dimensional scrolling grid
Two common GridView are Available:

1. Normal way, just display text in GridView layout.

2. Create a custom adapter to display image and text in GridView layout.

1. Normal GridView example

Display characters from A to Z in GridView layout. Quite simple

Step1:

Navigate to res- > layout- > Create New Layout and name it as "main.xml" and the content as shown below

Step2: Navigate to java file named as as shown below

Src->your application package-> and create a "GridViewActivity.java" and add the content
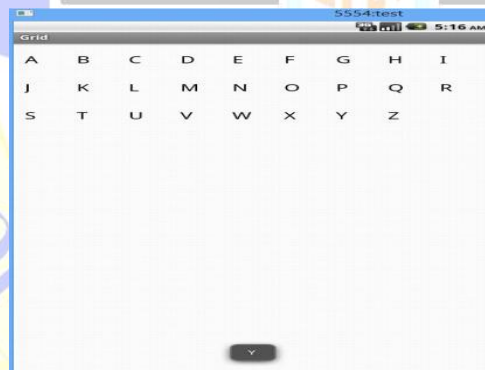package com.example.grid;

Step 3:
UpdateAndroidManifest.xml file contents
android:name="com.example.grid.GridViewActivity"
Step 4:
Run the project by right click the "GridViewActivity.java" file
Runas☐ Reconfiguration☐and Browse the project and click run

**Input & Output**



**Conclusion**
Thus the student can get the knowledge about the layouts implementation in android based
mobile devices. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains
POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

**Viva voce questions**

1)      What are the different layouts in android programming?

2)      Define different types Layouts

3)      List some types layout attributes?

### 13.     Write an android application to convert temperature from Celsius to Fahrenheit.

**Aim:**  To **write** a n android application program that converts from Celsius to Fahrenheit.

### Procedure

1.)  Create a new project by File-> New -> Android Project name it
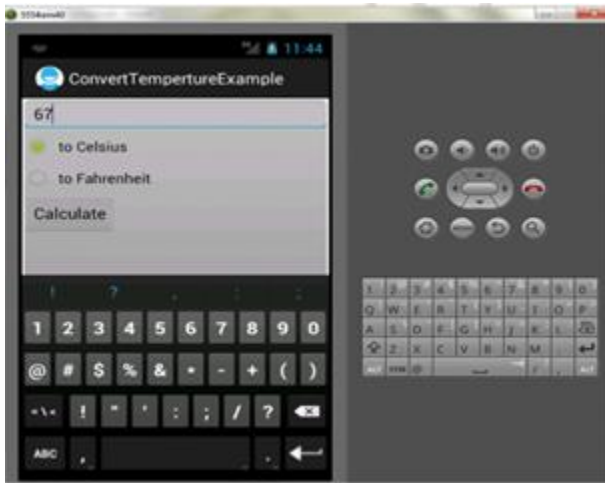
ConvertTemperatureExample.
3)

5.) Run for output.
Steps:

**Conclusion:** Student can get the knowledge about conversion of temperature to Fahrenheit and vice versa using android programming. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

Viva Voce Questions

1) What are the four essential states of an activity?

2) What is ANR?

3) When is the on Stop() method invoked?

4) What is the Android Manifest.xml?

**Aim:**  To write an android application program that demonstrates intents in mobile application development.
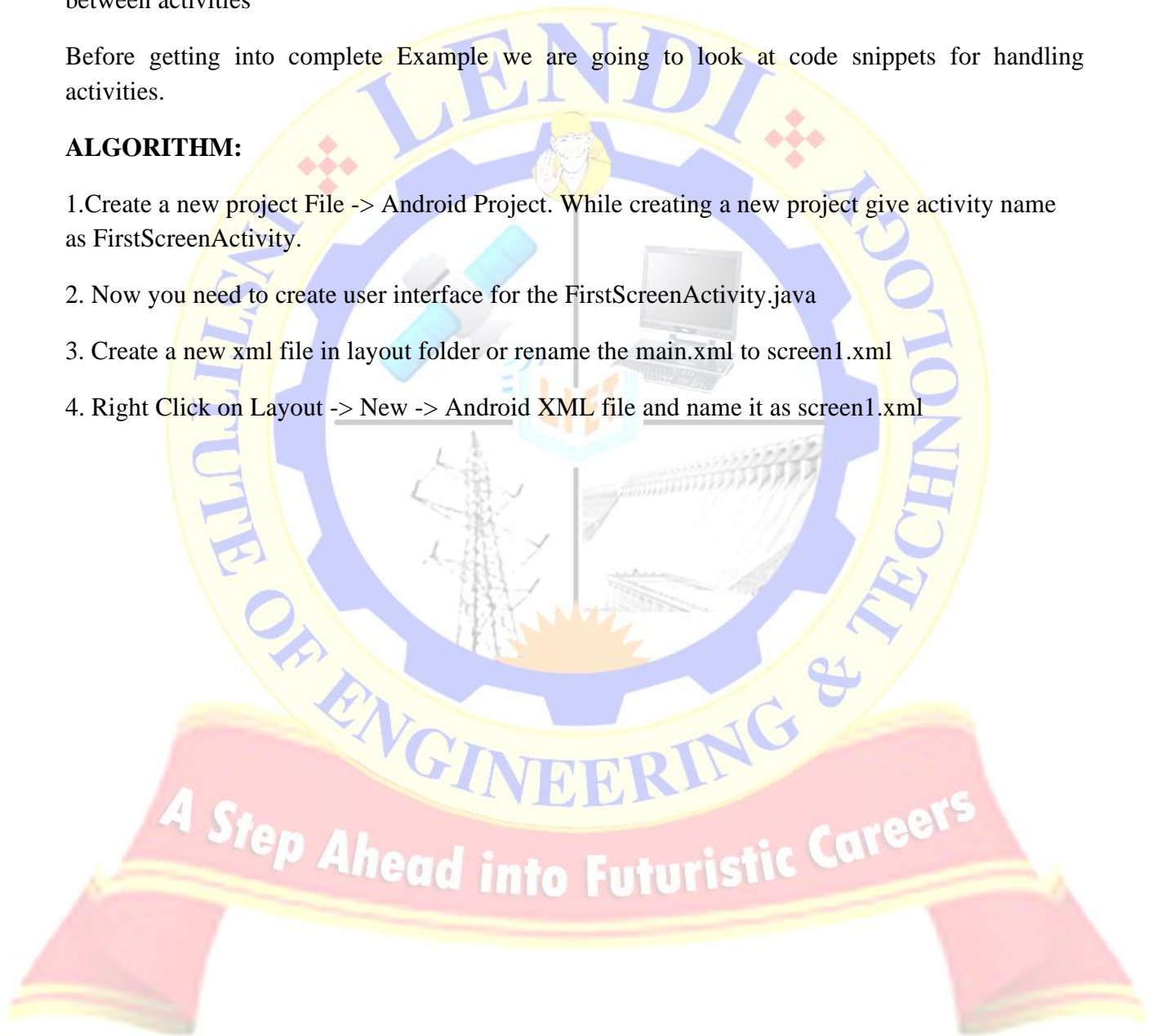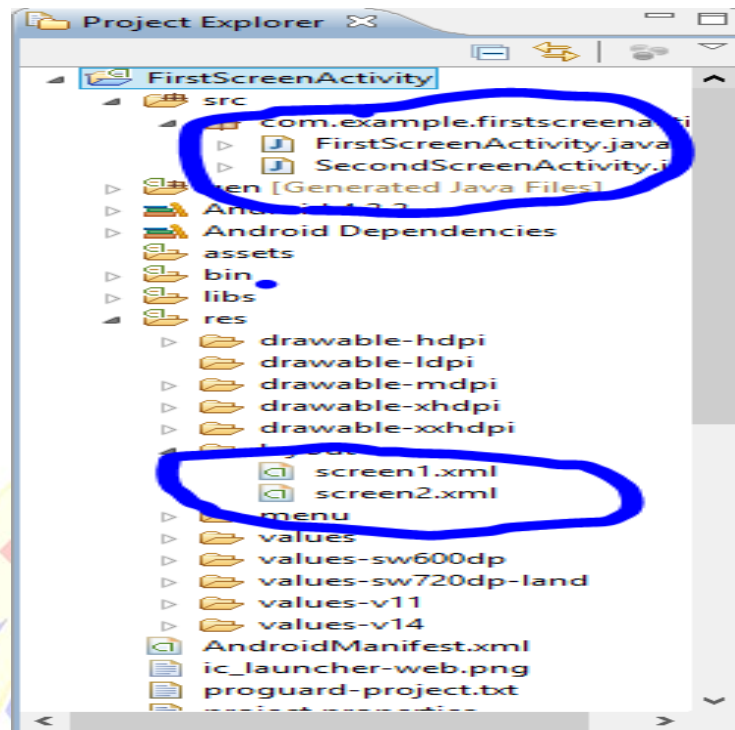
**Intent**

in Android user interface is displayed through an activity. In Android app development you might face situations where you need to switch between one Activity (Screen/View) to another. In this Example we will be discussing about switching between one Activity to another and sending data between activities

Before getting into complete Example we are going to look at code snippets for handling activities.

**ALGORITHM:**

1.Create a new project File -> Android Project. While creating a new project give activity name as FirstScreenActivity.

2. Now you need to create user interface for the FirstScreenActivity.java

3. Create a new xml file in layout folder or rename the main.xml to screen1.xml

4. Right Click on Layout -> New -> Android XML file and name it as screen1.xml

**4. Now insert the following code in screen1.xml to design a small layout. This layout contains simple form with a button.**

5. Now open your FirstScreenActivity.java and Type the following code. In the following code we are creating a new Intent and passing parameters on clicking button.

6. Create a class called SecondScreenActivity.java. Right Click on src/yourpackagefolder -> New -> Class and name it asSecondScreenActivity.java
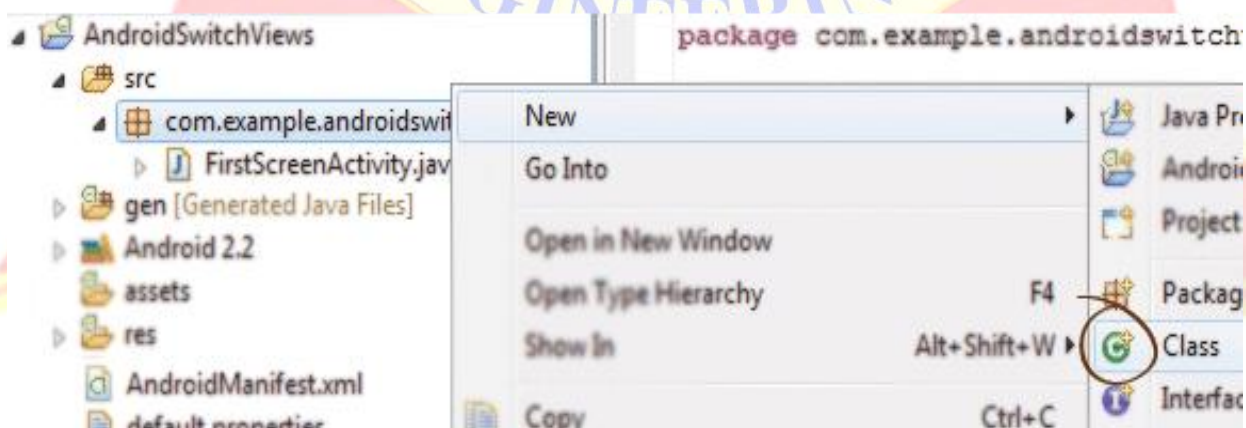
### Input & Output
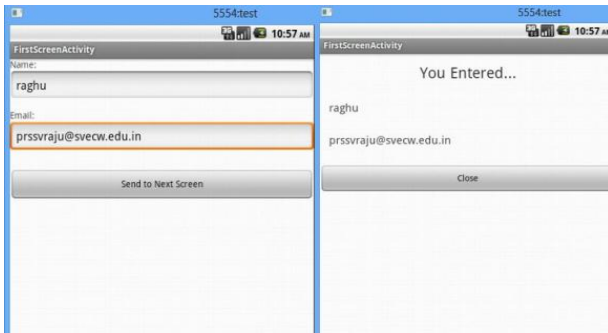


### Conclusion

Thus the students can get the knowledge about the intents using in android programming. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

### Viva Voce Questions

1) What are Intents?

2) Differentiate Activities from Services.

3) What is the importance of Default Resources?

4) What is AIDL?

4) What data types are supported by AIDL?

5) What is a visible activity?

6) How do you remove icons and widgets from the main screen of the Android device?

7) ) What are the core components under the Android application architecture?

8) What composes a typical Android application project?

9) What language is supported by Android for application development?

## EXERCISE - 14

**AIM: To Implement An Application That Writes Data To The Sd Card.**

### ALGORITM:

1) Open eclipse or android studio and select new android project
2) Give project name and select next
3) Choose the android version. Choose the lowest android version (Android 2.2) and select next
4) Enter the package name. Package name must be two word separated by comma and click finish
5) Go to package explorer in the left hand side. Select our project.
6) Go to res folder and select layout. Double click the main.xml file. Add the code below

### Input & Output:
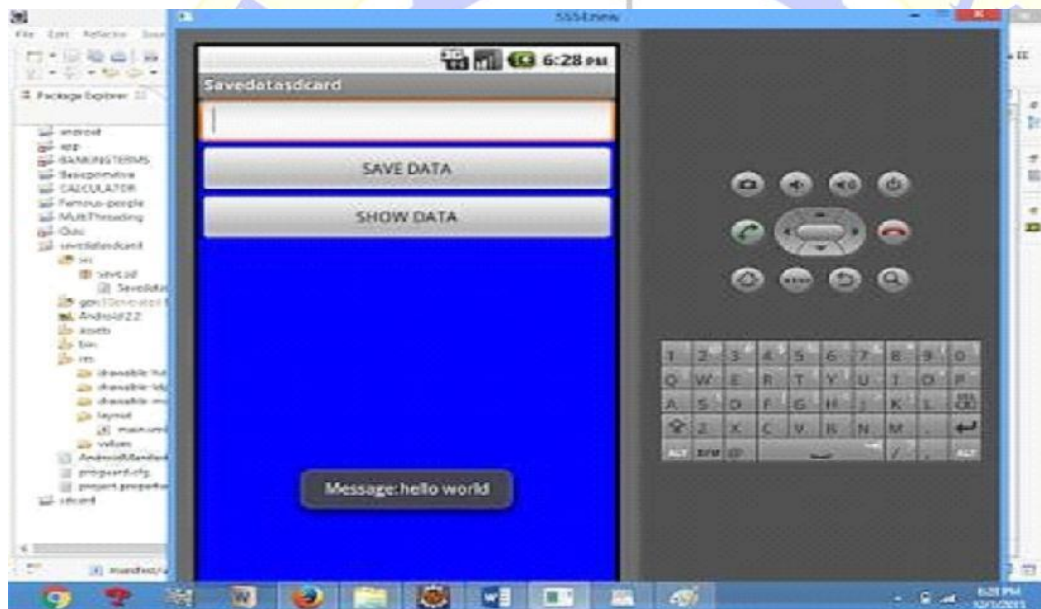


**Conclusion:** Thus the student can able to learn the procedure and implementation of writing the data in to SD card using Android programming. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

### Viva voce Questions

1) What is the use of an activity Creator?
2) Enumerate the steps in creating a bounded service through AIDL.
3) What is the importance of Default Resources?

## EXERCISE - 15
### AIM: To Develop an Android Native Application That Uses GPS Location Information

## ALGORITHM

1) Open eclipse and select new android project

2) Give project name and select next

3) Choose the android version. Choose the lowest android version (Android 2.2) and select next

4) Enter the package name. Package name must be two word separated by comma and click Finish

5) Go to package explorer in the left hand side. select our project.

6) Go to res folder and select layout. Double click the main.xml file. Add the code below

7) Now select mainactivity.java file and add code.

8) Go to src folder and Right Click on your package folder and choose new class and give the class names as GPS trace
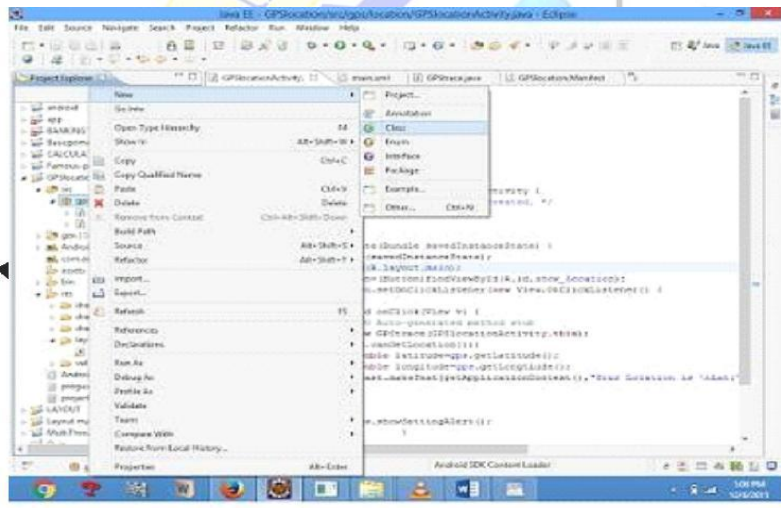


9) Select the GPStrace.java file and paste the code.

10) Go to manifest.xml file and add the code below

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

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

11) Now go to main.xml and right click .select run as option and select run Configuration

12) Android output is present in the android emulator as shown in below.

**Input & Output**
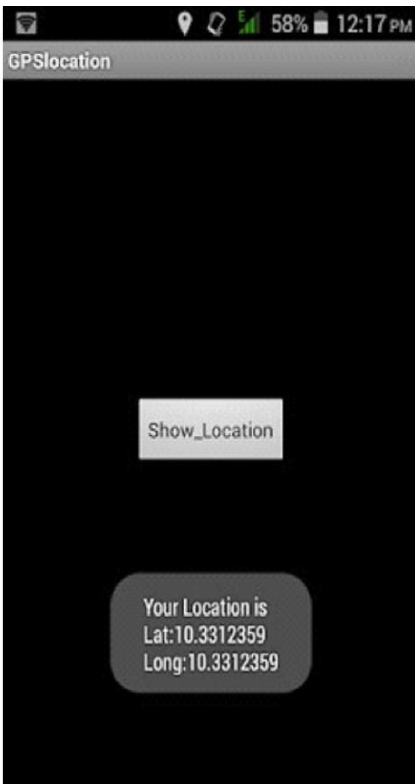


**Conclusion**: thus the student can know the Global Positioning System (GPS) location information using Android programming. This experiment maps with CO1, CO2, CO3, CO4 and the students can attains POS: PO1, PO2, PO3, PO5 & PSOS: PSO1

1) What is GPS?
2) What is native Application?
3) How to Add GPS to your Device?