Assignment No	9(Group B-3)
Title	A mobile application needs to be designed for using a Calculator (+, -,*, /, Sin, Cos, sq-root) with Memory Save/Recall using Extended precision floating point number format. Give the Required modeling, Design and Positive-Negative test cases.
Date	
Signature	

Assignment No: 9

1. TITLE

A mobile application needs to be designed for using a Calculator (+, - ,*, /, Sin, Cos, sq-root) with Memory Save/Recall using Extended precision floating point number format. Give the Required modeling, Design and Positive-Negative test cases.

2. PREREQUISITES

- 64-bit Fedora or equivalent OS with 64-bit Intel-i5/i7
- Java 1.7.0
- Android Studio

3. OBJECTIVE

- To learn the Android Studio.
- To study the design and implementation of mobile application for calculator.

4. THEORY

Android Studio Overview

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine
- And much more

Android Project Structure

By default, Android Studio displays your project files in the *Android* project view. This view shows a flattened version of your project's structure that provides quick access to the key source files of Android projects and helps you work with the Gradle-based build system. The *Android* project view:

- Shows the most important source directories at the top level of the module hierarchy.
- Groups the build files for all modules in a common folder.
- Groups all the manifest files for each module in a common folder.
- Shows resource files from all Gradle source sets.
- Groups resource files for different locales, orientations, and screen types in a single group per resource type

java/ - Source files for the module.
manifests/ - Manifest files for the module.
res/ - Resource files for the module.
Gradle Scripts/ - Gradle build and property files.

Positive Testing:

Test Case ID	Expected Result	Actual Result	Status
1	Check if all the numbers are working (0 to 9)	All the numbers are working (0 to 9)	
2	Check if the arithmetic keys (+, -, *, %, /) are working.	Arithmetic keys (+, -, *, %, /) are working	
3	Check if the bracket keys are working.	The bracket keys are working	
4	Check if the sum or equal key is working.	Sum or equal key is working properly	
5	Check the addition of two integer numbers.	Addition of two integers	
6	Check the addition of two negative numbers	Addition of two negative numbers	
7	Check the subtraction of two integer numbers.	Subtraction of two integers	
8	Check the subtraction of two negative numbers	Subtraction of two negative numbers	
9	Check the multiplication of two integer numbers	Multiplication of two integers	
10	Check the division of two integer numbers	Division of two integers	

Negative Testing:

Test Case ID	Expected Result	Actual Result	Status
1	Check if it is allowing letters instead of numbers	It is taking only numbers as input	

5. MATHEMATICAL MODEL:

Let, S be the System Such that,
A={ S, E, I,O, F, DD, NDD, success, failure }
Where,
S= Start state,
E= End State,
I= Set of Input
O= Set of Out put
F =Set of Function

DD=Deterministic Data
NDD=Non Deterministic Data
Success Case: It is the case when all the inputs are given by system are entered correctly.
Failure Case: It is the case when the input does not match the validation Criteria.

6. APPLICATION FLOW

Steps	Description
1	You can use Android studio IDE/android-adt-bundle to create an Android application under a package com.example.calci.myapplication;. While creating this project, make sure you Target SDK and Compile With at the latest version of Android SDK to use higher levels of APIs.
2	Modify src/MainActivity.java file to add Calculator code.
3	Modify the res/layout/activity_main to add respective XML components
4	Create a new folder under Calculator
5	Run the application and choose a running android device and install the application on it and verify the results

7. CONCLUSION

A mobile application is designed for a Calculator (+, - ,*, /, Sin, Cos, sq-root) with Memory Save/Recall using Extended precision floating point number format.

8. REFERENCES

[1] HASSAN GOMAA, Software Modeling and Design, Cambridge university Press, 2011, ISBN-13 978-1-107-44735-6.
[2] Erich Gamma, Richard Helm, Ralph Johnson , John Vlissides, Design patterns Elements of Reusable Object-Oriented Software.
[3] Srinivasan Desikan, "Software Testing Principals and practices", Pearson Publication ISBN-13 978-8-17-758295-6.

9. FAQ’s

1. Which is the latest Android OS ?
2. Which are the types of Layouts ?
3. What is positive and negative testing ?