Experiment 10: Implementation of income tax/loan EMI calculator and deploy the same on real devices

PART A

A.1 Aim: To Implementation of income tax/loan EMI calculator and deploy the same on real devices.

A.2 Objectives: **To introduce students with various tools like** Android Studio, NS2, Wireshark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome: **After successful completion of this experiment students will be able to** Implementation of income tax/loan EMI calculator and deploy the same on real devices

A.4 Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

15. Open android studio and select new android project .
16. Give project name and select next

17. Choose the android version.
18. Enter the package name. package name must be two word separated by comma and click finish
19. Go to package explorer in the left hand side and select our project.
20. Go to res folder and select layout. Double click the main.xml file 21. Now you can see the Graphics layout window.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent" android:background="#D8D0D0"
tools:context=".MainActivity">

<EditText
android:id="@+id/editText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp" android:ems="10"
android:hint="@string/enter_principle"
android:inputType="number"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.45"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.203" android:autofillHints=""
/>

<EditText android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:ems="10"
android:hint="@string/enter_time"
android:inputType="number"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.45"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.274"
android:autofillHints="" />
```

```
<EditText android:id="@+id/editText3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:ems="10"
android:hint="@string/enter_rate_of_interest" android:inputType="number"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.45"
app:layout_constraintStart_toStartOf="parent"
```

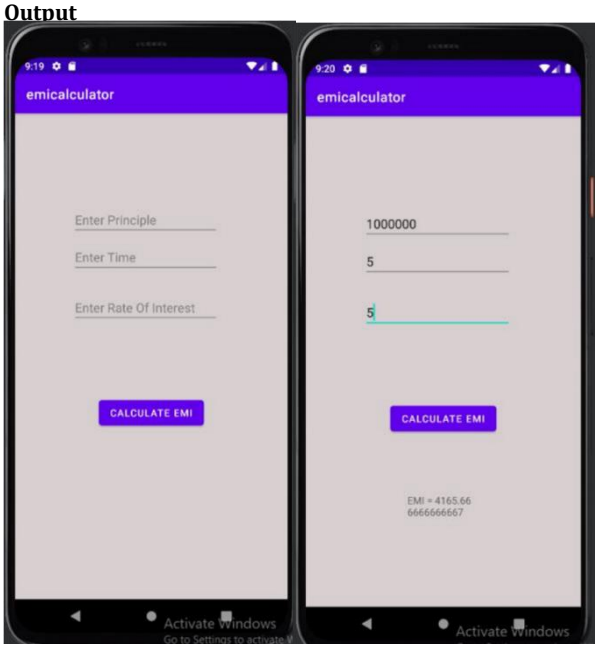
```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.386"
android:autofillHints="" />

<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:text="@string/calculate_emi"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.62" />

<TextView
android:id="@+id/textView"
android:layout_width="95dp"
android:layout_height="104dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.486"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.901" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.emicalculator;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity  {
   Button calculate;
TextView result;
EditText pe, ne, re;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

      calculate =  findViewById(R.id.button);
      result =  findViewById(R.id.textView);
      pe = findViewById(R.id.editText);
      ne = findViewById(R.id.editText2);
      re = findViewById(R.id.editText3);
calculate.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Double p = Double.parseDouble(pe.getText().toString());
            Double n = Double.parseDouble(ne.getText().toString());
            Double r = Double.parseDouble(re.getText().toString());
```

```
            Double rate= (r/1200);
            Double res = (p * rate * Math.pow(1 + rate, n) / Math.pow(1 + rate, n) - 1);
result.setText("EMI = "+String.valueOf(res));
          }
       });
    }
}
```

Output



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE B Comps	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exp10;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```
import java.text.DecimalFormat;
public class MainActivity extends AppCompatActivity {
private EditText loanAmountEditText;
private EditText interestRateEditText;
private EditText loanTenureEditText;
private TextView resultTextView;
@SuppressLint("SetTextI18n")
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
loanAmountEditText = findViewById(R.id.loanAmountEditText);
interestRateEditText = findViewById(R.id.interestRateEditText);
loanTenureEditText = findViewById(R.id.loanTenureEditText);
Button calculateButton = findViewById(R.id.calculateButton);
resultTextView = findViewById(R.id.resultTextView);
calculateButton.setOnClickListener(view -> {
try {
// Read input values
double loanAmount =
Double.parseDouble(loanAmountEditText.getText().toString().trim());
double interestRate =
Double.parseDouble(interestRateEditText.getText().toString().trim());
int loanTenure = Integer.parseInt(loanTenureEditText.getText().toString().trim());
// Validate input values
if (loanAmount <= 0 || loanTenure <= 0) {
resultTextView.setText("Please enter valid loan amount and tenure.");
return;
}
double monthlyInterestRate = interestRate / 1200;
int numberOfPayments = loanTenure * 12;
double emi;
if (interestRate == 0) {
// Simple division if no interest
emi = loanAmount / numberOfPayments;
} else {
```

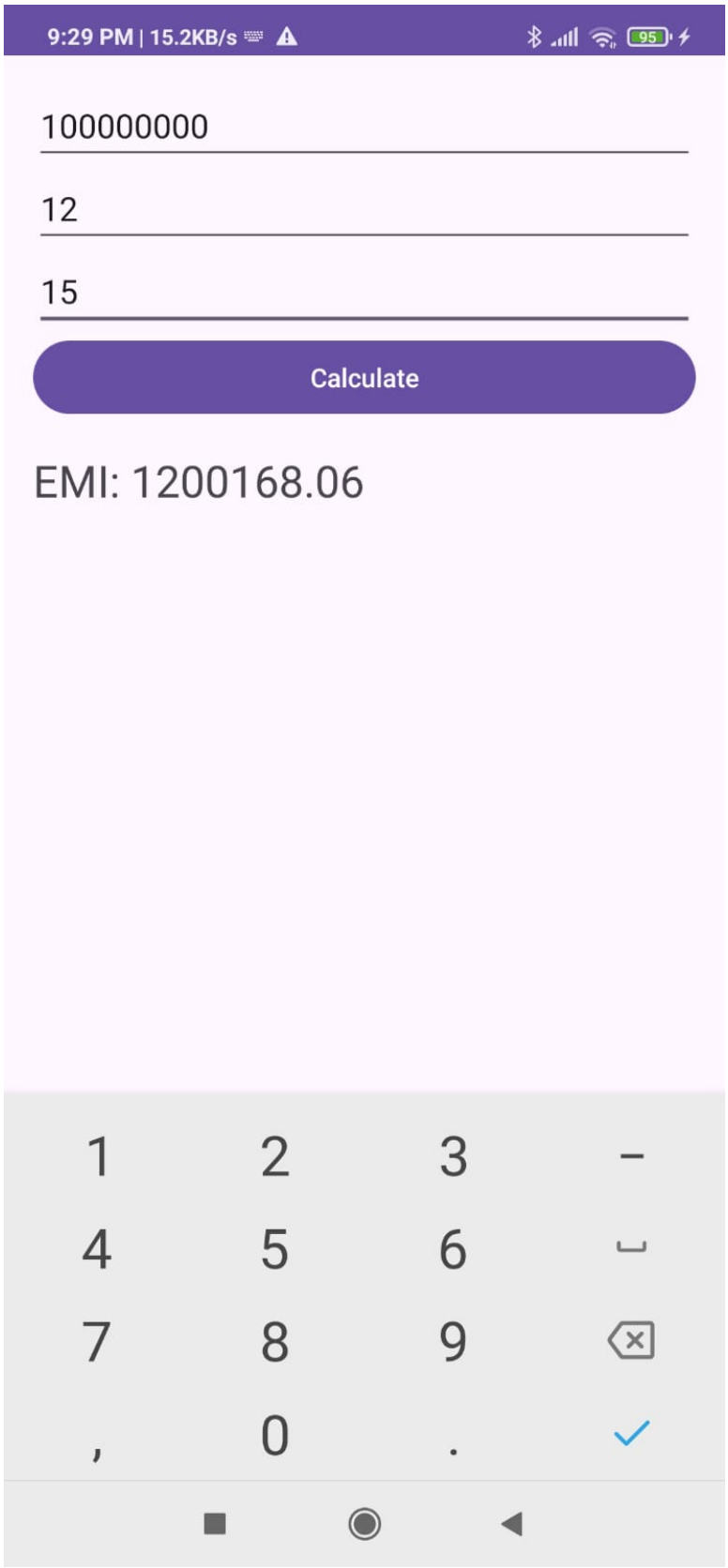
```
// EMI formula
emi = (loanAmount * monthlyInterestRate * Math.pow(1 + monthlyInterestRate,
numberOfPayments))
/ (Math.pow(1 + monthlyInterestRate, numberOfPayments) - 1);
}
DecimalFormat decimalFormat = new DecimalFormat("#.##");
resultTextView.setText("EMI: " + decimalFormat.format(emi));
} catch (NumberFormatException e) {
resultTextView.setText("Invalid input! Please enter valid numbers.");
}
});
}
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools">
<application
android:allowBackup="true"
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.Exp10"
tools:targetApi="31">
<activity
android:name=".MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
```

</manifest>

B.2 Input and Output:


9:29 PM | 15.2KB/s
100000000
12
15
Calculate
EMI: 1200168.06

B.3 Observations and learning:

In this experiment, an Android application was successfully developed to calculate loan EMI based on user inputs for principal amount, interest rate, and loan tenure. The implementation used Android Studio, Java, and XML to design the user interface and perform EMI calculations. The app included input fields for entering financial data, a button to trigger the calculation, and a TextView to display the EMI result. The experiment demonstrated how to handle user input, perform mathematical computations using Java, and update UI elements dynamically. Additionally, the application was tested on real devices to ensure proper functionality

B.4 Conclusion:

This experiment provided valuable hands-on experience in developing a functional Android application for financial calculations. By working with UI components, event handling, and mathematical operations, students gained practical knowledge of mobile app development. Deploying the application on real devices helped in understanding real-world constraints such as screen responsiveness and user experience. The experiment reinforced the importance of software tools like Android Studio and Java for developing interactive applications. This knowledge can be extended to build more complex financial applications, making it a useful foundation for real-world software development.

B.5 Question of Curiosity

Explain different steps required to build up this project?

1.Create a new project in Android Studio with an empty activity.
2. Open the activity_main.xml file and design the user interface for the loan calculator. Thiscan include EditText views for the loan amount, interest rate, and loan term, as well as a TextView for the calculated monthly payment.
3. In the MainActivity.java file, declare variables for the loan amount, interest rate, loan term, and monthly payment. Also, create references to the EditText and TextView views.
4. Implement a listener for the calculate button that extracts the values from the EditText views, calculates the monthly payment using the loan formula, and displays the result in theTextView.
5. Use the following formula to calculate the monthly payment:
Monthly payment = (loan amount * interest rate / 1200) / (1 - (1 + interest rate / 1200) ^ (-loan term))
6. Run the app to test the loan calculator