NAME:Bhavik Ransubhe

CLASS: TE (B) COMP

ROLL NO : 39055

**Android: Design a mobile app to store data using internal or external storage .**

**--------------------------------------------------------------------------------------------**

**Aim:** Design a mobile app to store data using internal or external storage. (**Internal done**)

**Learning Objective**:

To study and implementation of mobile application for media player

**Software Requiremenrts:**

• Android Studio

• Windows 10 64 bit

• Java JDK1.7 or later version

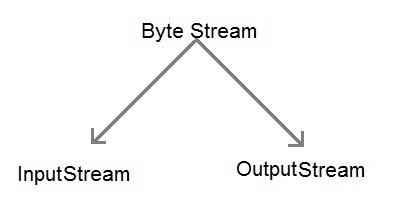
**Hardware Requirements:**

PC/Laptop with min 4GB RAM , 500GB HDD, Intelcore i5 Processor.

**Theory:**

**Java Byte Stream Classes:**

Byte stream is defined by using two abstract class at the top of hierarchy, they are InputStream and OutputStream.



These two abstract classes have several concrete classes that handle various devices such as disk files, network connection etc.

|  |  |
| --- | --- |
| **Stream class** | **Description** |
| **BufferedInputStream** | Used for Buffered Input Stream. |
| **BufferedOutputStream** | Used for Buffered Output Stream. |
| **DataInputStream** | Contains method for reading java standard datatype |
| **DataOutputStream** | An output stream that contain method for writing java standard data type |
| **FileInputStream** | Input stream that reads from a file |
| **FileOutputStream** | Output stream that write to a file. |
| **InputStream** | Abstract class that describe stream input. |
| **OutputStream** | Abstract class that describe stream output. |
| **PrintStream** | Output Stream that contain print() and println() method |

#### **Some important Byte stream classes.**

These classes define several key methods. Two most important are

1. read() : reads byte of data.
2. write() : Writes byte of data.

**Java FileInputStream class methods:**

|  |  |  |
| --- | --- | --- |
| **Method** | | **Description** |
| int available() | It is used to return the estimated number of bytes that can be read from the input stream. | |
| int read() | It is used to read the byte of data from the input stream. | |
| int read(byte[] b) | It is used to read up to **b.length** bytes of data from the input stream. | |
| int read(byte[] b, int off, int len) | It is used to read up to **len** bytes of data from the input stream. | |
| long skip(long x) | It is used to skip over and discards x bytes of data from the input stream. | |
| FileChannel getChannel() | It is used to return the unique FileChannel object associated with the file input stream. | |
| FileDescriptor getFD() | It is used to return the [FileDescriptor](https://www.javatpoint.com/java-filedescriptor-class) object. | |
| protected void finalize() | It is used to ensure that the close method is call when there is no more reference to the file input stream. | |
| void close() | It is used to closes the [stream](https://www.javatpoint.com/java-8-stream). | |

**FileOutputStream class methods**

|  |  |  |
| --- | --- | --- |
| **Method** | | **Description** |
| protected void finalize() | It is used to clean up the connection with the file output stream. | |
| void write(byte[] ary) | It is used to write **ary.length** bytes from the byte [array](https://www.javatpoint.com/array-in-java) to the file output stream. | |
| void write(byte[] ary, int off, int len) | It is used to write **len** bytes from the byte array starting at offset **off** to the file output stream. | |
| void write(int b) | It is used to write the specified byte to the file output stream. | |
| FileChannel getChannel() | It is used to return the file channel object associated with the file output stream. | |
| FileDescriptor getFD() | It is used to return the file descriptor associated with the stream. | |
| void close() | It is used to closes the file output stream. | |

**CODE:**

FILE 1: MainActivity.java:-

**package** com.br14x.storage;  
  
**import** android.content.Context;  
  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** java.io.BufferedReader;  
**import** java.io.FileNotFoundException;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 EditText **editTextFileName**,**editTextData**;  
 Button **saveButton**,**readButton**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **editTextFileName**=findViewById(R.id.***editText1***);  
 **editTextData**=findViewById(R.id.***editText2***);  
 **saveButton**=findViewById(R.id.***button1***);  
 **readButton**=findViewById(R.id.***button2***);  
  
 *//Performing Action on Read Button* **saveButton**.setOnClickListener(**new** View.OnClickListener(){  
  
 @Override  
 **public void** onClick(View arg0) {  
 String filename=**editTextFileName**.getText().toString();  
 String data=**editTextData**.getText().toString();  
  
 FileOutputStream fos;  
 **try** {  
 fos = openFileOutput(filename, Context.***MODE\_PRIVATE***);  
 *//default mode is PRIVATE, can be APPEND etc.* fos.write(data.getBytes());  
 fos.close();  
  
 Toast.*makeText*(getApplicationContext(),filename + **" saved"**,  
 Toast.***LENGTH\_LONG***).show();  
  
  
 } **catch** (FileNotFoundException e) {e.printStackTrace();}  
 **catch** (IOException e) {e.printStackTrace();}  
  
 }  
  
 });  
  
 *//Performing Action on Read Button* **readButton**.setOnClickListener(**new** View.OnClickListener(){  
  
 @Override  
 **public void** onClick(View arg0) {  
 String filename=**editTextFileName**.getText().toString();  
 StringBuffer stringBuffer = **new** StringBuffer();  
 **try** {  
 *//Attaching BufferedReader to the FileInputStream by the help of InputStreamReader* BufferedReader inputReader = **new** BufferedReader(**new** InputStreamReader(  
 openFileInput(filename)));  
 String inputString;  
 *//Reading data line by line and storing it into the stringbuffer* **while** ((inputString = inputReader.readLine()) != **null**) {  
 stringBuffer.append(inputString + **"\n"**);  
 }  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 *//Displaying data on the toast* Toast.*makeText*(getApplicationContext(),stringBuffer.toString(),Toast.***LENGTH\_LONG***).show();  
  
 }  
  
 });  
 }  
}

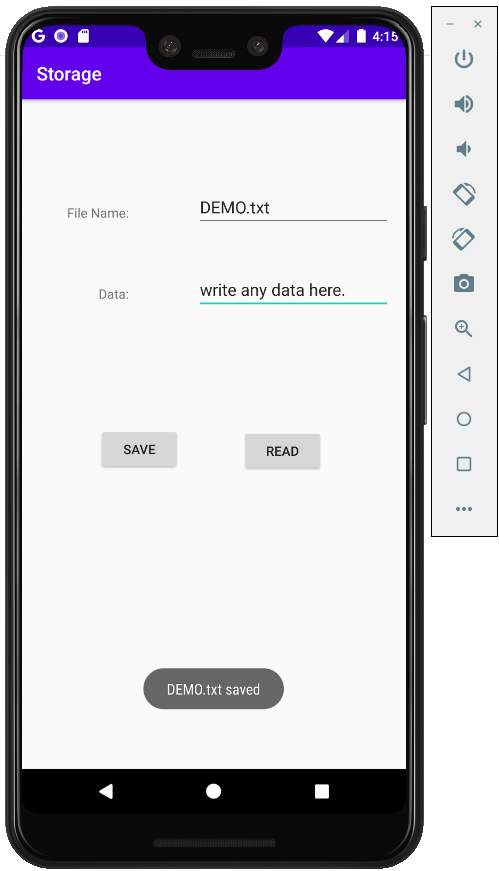
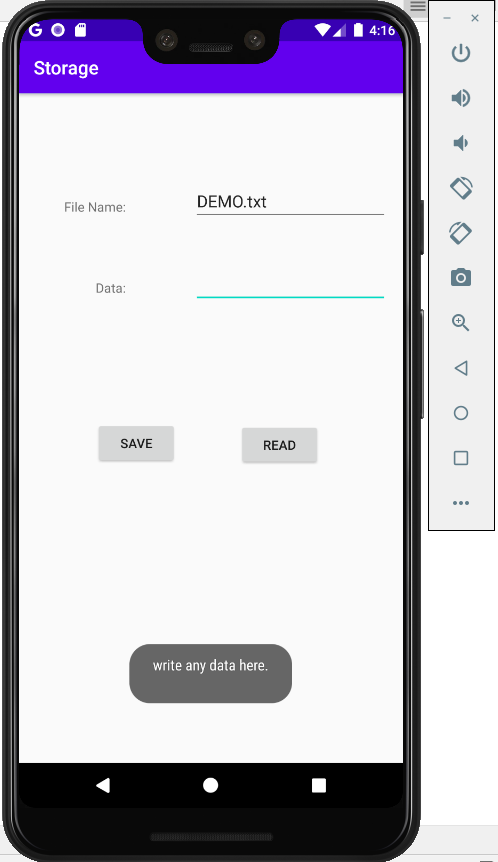
FILE 2:activity\_main.xml:-

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns: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"  
 tools:context=".MainActivity"**>  
  
 <**EditText  
 android:id="@+id/editText1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginTop="93dp"  
 android:layout\_marginEnd="16dp"  
 android:ems="10"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
  
 <**requestFocus** />  
 </**EditText**>  
  
 <**EditText  
 android:id="@+id/editText2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editText1"  
 android:layout\_alignRight="@+id/editText1"  
 android:layout\_marginTop="43dp"  
 android:layout\_marginEnd="16dp"  
 android:ems="10"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText1"** />  
  
 <**TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBaseline="@+id/editText1"  
 android:layout\_alignBottom="@+id/editText1"  
 android:layout\_alignParentLeft="true"  
 android:layout\_marginTop="112dp"  
 android:layout\_marginEnd="72dp"  
 android:text="File Name:"  
 app:layout\_constraintEnd\_toStartOf="@+id/editText1"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBaseline="@+id/editText2"  
 android:layout\_alignBottom="@+id/editText2"  
 android:layout\_alignParentLeft="true"  
 android:layout\_marginTop="68dp"  
 android:layout\_marginEnd="72dp"  
 android:text="Data:"  
 app:layout\_constraintEnd\_toStartOf="@+id/editText2"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView1"** />  
  
 <**Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editText2"  
 android:layout\_alignLeft="@+id/editText2"  
 android:layout\_marginStart="82dp"  
 android:layout\_marginTop="133dp"  
 android:text="save"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"** />  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBaseline="@+id/button1"  
 android:layout\_alignBottom="@+id/button1"  
 android:layout\_marginTop="126dp"  
 android:layout\_marginEnd="88dp"  
 android:layout\_toRightOf="@+id/button1"  
 android:text="read"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText2"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**OUTPUT:**

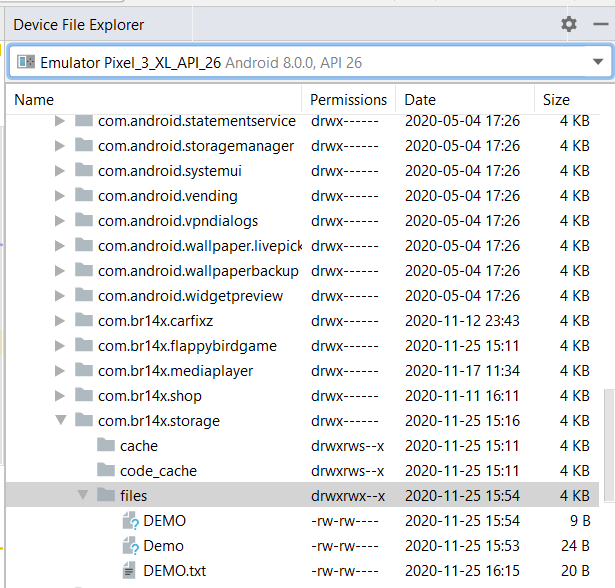
*1)Saving a file with file name and data in it :-*

2)Write file name and press READ button to see data present in it ,in TOAST form :-

1.**** 2. 

Files saved Can be seen in following path :

*View->Tool Windows -> Device File Explorer->data->android->packagename:-*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*