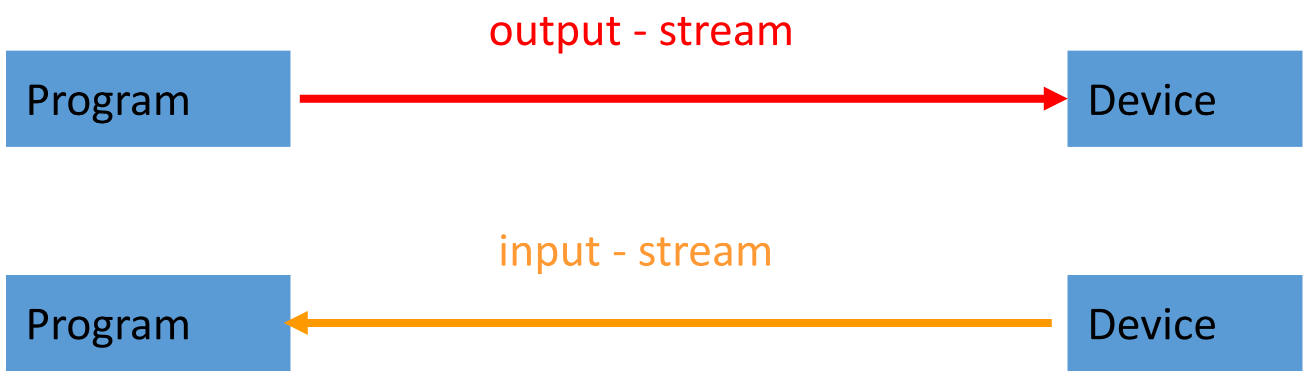
Android I/O

Overview of I/O

- Usual Purpose: storing data to **nonvolatile** devices, e.g. Harddisk
- Class provided by package java.io
  - http://developer.android.com/reference/java/io/package-summary.html
- Data is transferred to devices by **streams**



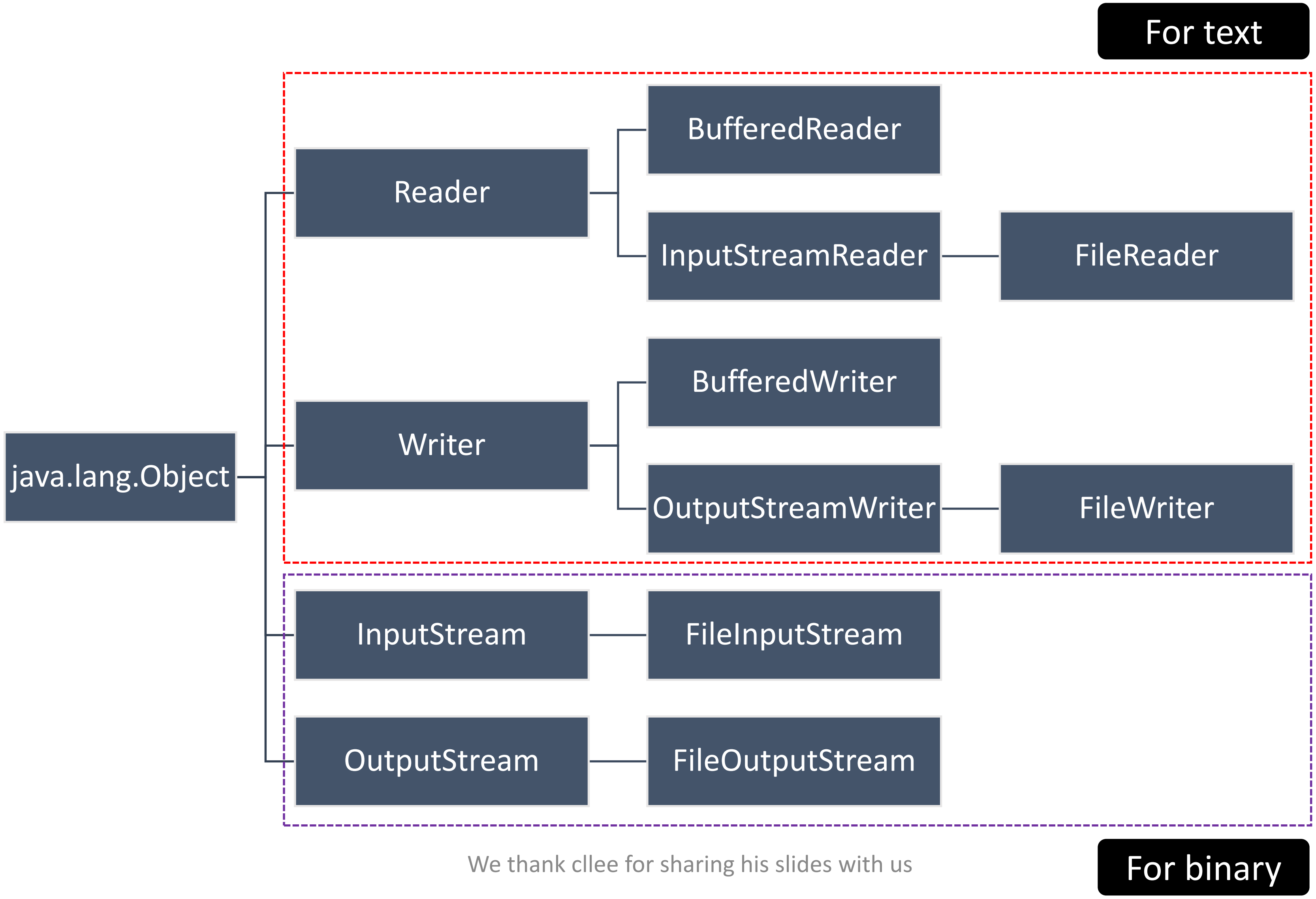
We thank cllee for sharing his slides with us

Streams

- Java distinguishes between 2 types of streams:
  - Text
  - Binary streams

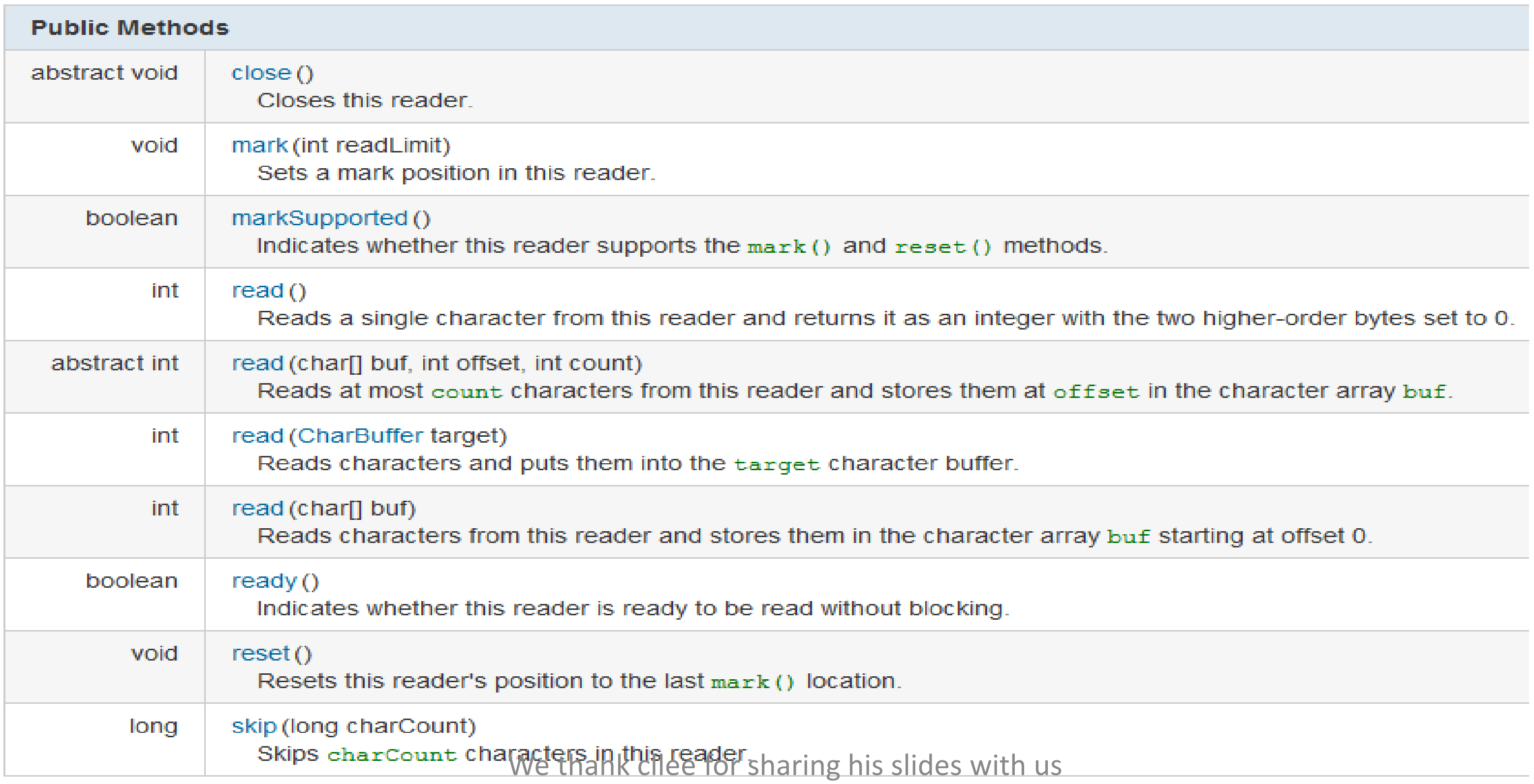
- Results in 4 base-classes dealing with I/O:
  - Reader: text-input
  - Writer: text-output
  - InputStream: byte-input
  - OutputStream: byte-output

Streams


For text
java.lang.Object
Reader
BufferedReader
InputStreamReader
FileReader
Writer
BufferedWriter
OutputStreamWriter
FileWriter
InputStream
FileInputStream
OutputStream
FileOutputStream
For binary

Reading Textfiles

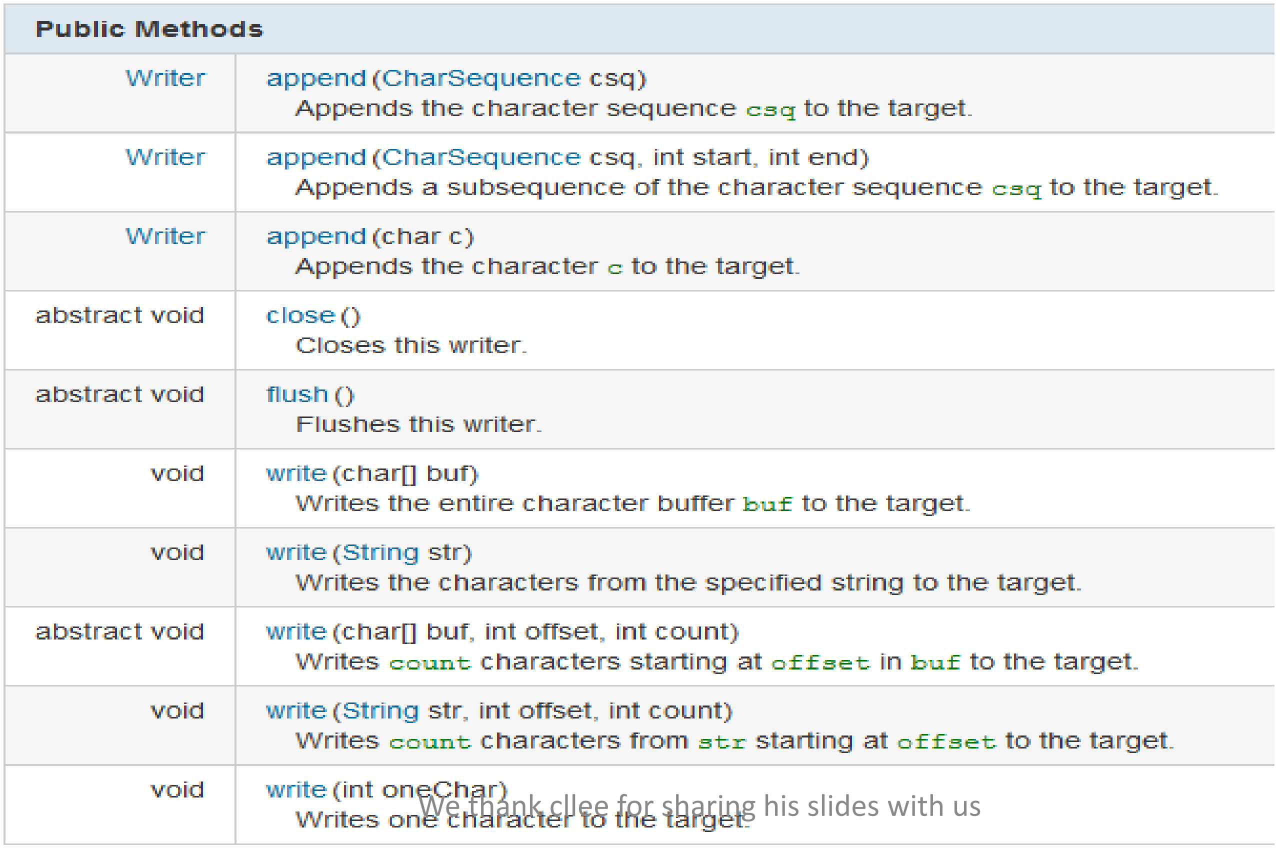
- Reader (http://developer.android.com/reference/java/io/Reader.html)
  - FileReader
  - BufferReader



Public Methods	
abstract void	close () Closes this reader.
void	mark (int readLimit) Sets a mark position in this reader.
boolean	markSupported () Indicates whether this reader supports the `mark()` and `reset()` methods.
int	read () Reads a single character from this reader and returns it as an integer with the two higher-order bytes set to 0.
abstract int	read (char[] buf, int offset, int count) Reads at most `count` characters from this reader and stores them at `offset` in the character array `buf`.
int	read (CharBuffer target) Reads characters and puts them into the `target` character buffer.
int	read (char[] buf) Reads characters from this reader and stores them in the character array `buf` starting at offset 0.
boolean	ready () Indicates whether this reader is ready to be read without blocking.
void	reset () Resets this reader's position to the last `mark()` location.
long	skip (long charCount) Skips `charCount` characters in this reader.

Writing Textfiles

- Writer (http://developer.android.com/reference/java/io/Writer.html)
  - FileWriter
  - BufferWriter



Public Methods	
Writer	append (CharSequence csq) Appends the character sequence `csq` to the target.
Writer	append (CharSequence csq, int start, int end) Appends a subsequence of the character sequence `csq` to the target.
Writer	append (char c) Appends the character `c` to the target.
abstract void	close () Closes this writer.
abstract void	flush () Flushes this writer.
void	write (char[] buf) Writes the entire character buffer `buf` to the target.
void	write (String str) Writes the characters from the specified string to the target.
abstract void	write (char[] buf, int offset, int count) Writes `count` characters starting at `offset` in `buf` to the target.
void	write (String str, int offset, int count) Writes `count` characters from `str` starting at `offset` to the target.
void	write (int oneChar) Writes one character to the target.

Read/Write Textfiles

- FileReader

```
FileReader fr = new FileReader(“PATH_NAME");
```

- FileWriter

```
FileWriter wr = new FileWriter(“PATH_NAME");
```

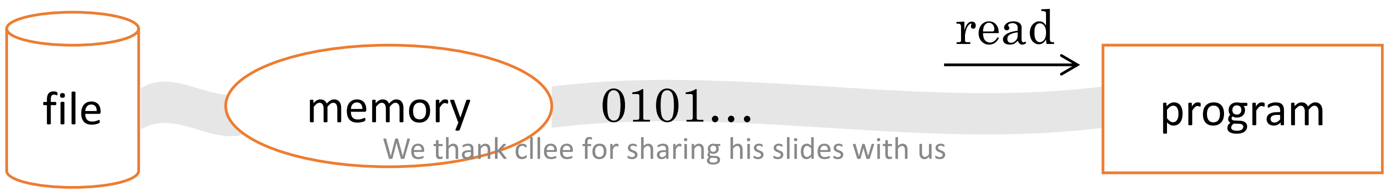
- Example:

```
FileReader fr = new FileReader("in.txt");
FileWriter wr = new FileWriter("out.txt");

int num;
while(( num = fr.read()) != -1 )
       wr.write(num);

fr.close();
wr.close();
```

Read/Write Textfiles

- Using FileWriter/FileReader
  - is not very convenient (only **String-output** possible)
  - is not efficient (**every character is written in a single step**, invoking a huge overhead)

- Better: wrap FileWriter/FileReader with processing streams
  - BufferedWriter/BufferReader
  - BufferStream means streams not stored in disk; otherwise, **stored in memory first** and readout after program needed.



Read/Write Textfiles

- BufferedReader

```
FileReader in = new FileReader("test.txt");
BufferedReader inf = new BufferedReader(in);
```

- BufferedWriter

```
FileWriter out = new FileWriter("test.txt");
BufferedWriter b = new BufferedWriter(out);
```

- Example:

```
BufferedReader inf = new BufferedReader(new FileReader("in.txt"));
BufferedWriter outf = new BufferedWriter(new FileWriter("out.txt"));

int num;
while(( num = inf.read()) != -1 )
        outf.write(num);

inf.close();
outf.close();
```

Binary Files

- Stores binary images of information identical to the binary images stored in main memory
- Binary files are more efficient in terms of processing time space utilization
- Class:
  - FileInputStream
  - FileOutputStream
  - DataInputStream
  - DataOutputStream

Read/Write Binary Files

- Example

```
DataInputStream inf = new DataInputStream(
                      new FileInputStream("xxx.mp3"));
DataOutputStream outf = new DataOutputStream(
                      new FileOutputStream("xxx2.mp3"));

int num;
while ((num = inf.read()) != -1)
       outf.write(num);

inf.close();
outf.close();
```

Android for read/write

- Store/Load Externel Storage
  - Using getExternalStorageState() to check SDCARD

```
String state = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(state)) {
    //can W/R sdcard
}
else if
        (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)){
    //Only read
}
else {
    //cannot use sdcard
}
```

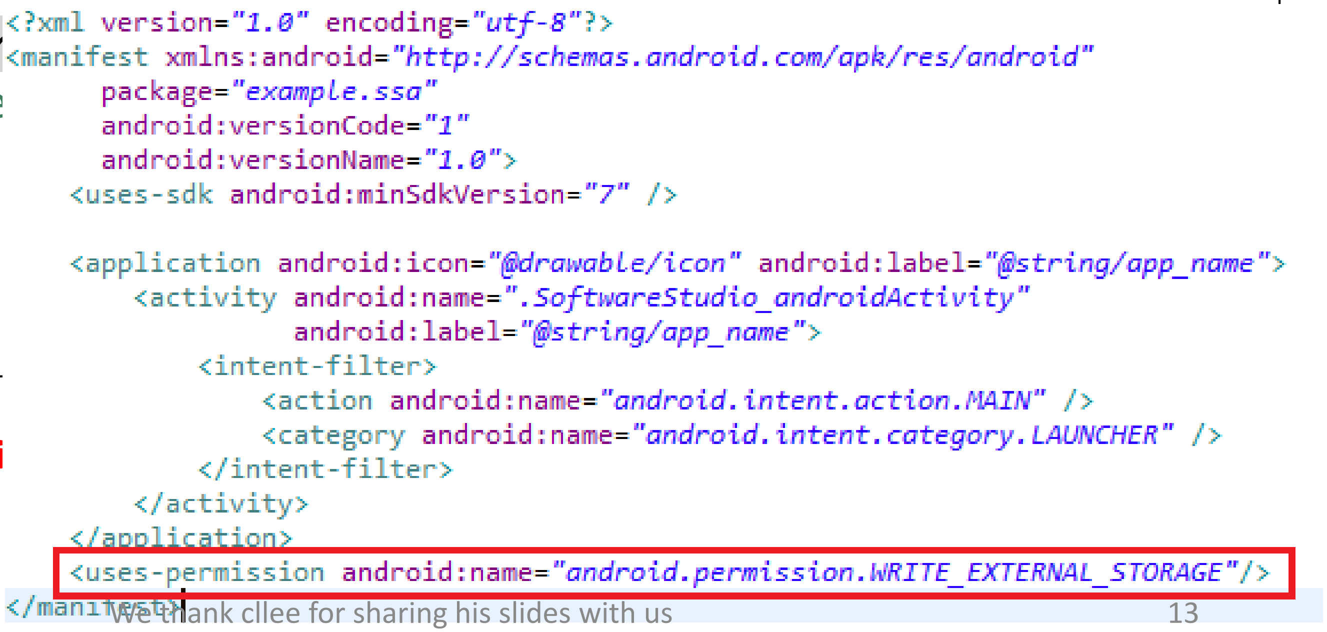
  - Permission of AndroidManifest.xml
    - **<uses-permission android:name="*android.permission.WRITE_EXTERNAL_STORAGE*"/>**

Android for read/write

- Store/Load Externel Storage
  - Using getExternalStorageState() to check SDCARD

```
String state = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(state)) {
    //can W/R sdcard
}
else if
        (Envir
    //Only re
}
else {
    //cannot
}
```

- Permission of
  - <uses-permissi



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="example.ssa"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="7" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".SoftwareStudio_androidActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
</manifest>
```

Android for read/write

- Load/Write File
  - API LEVEL 8
    - **Context.getExternalFilesDir(String type)**

```
File path=context.getExternalFilesDir(null);
File file=new File(path,“xxx.jpg");
```

  - Below API LEVEL 7
    - **Environment.getExternalStorageDirectory()**

```
File path=Environment.getExternalStorageDirectory();
File file=new File(path,“xxx.jpg");
```

Android for read/write

- Example

```
String state = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(state)) {
        sdDir = context.getExternalFilesDir(null);
        FileInputStream fis = new FileInputStream(sdDir+FILENAME);
        DataInputStream inf = new DataInputStream(fis);
        FileOutputStream fos = new
                FileOutputStream(sdDir+FILENAME+"_copy");
        DataOutputStream outf = new DataOutputStream(fos);
        int num;

        while ((num = inf.read()) != -1) {
                outf.write(num);
        }

        inf.close();
        outf.close();

}
else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {}
else {}
```

Your time

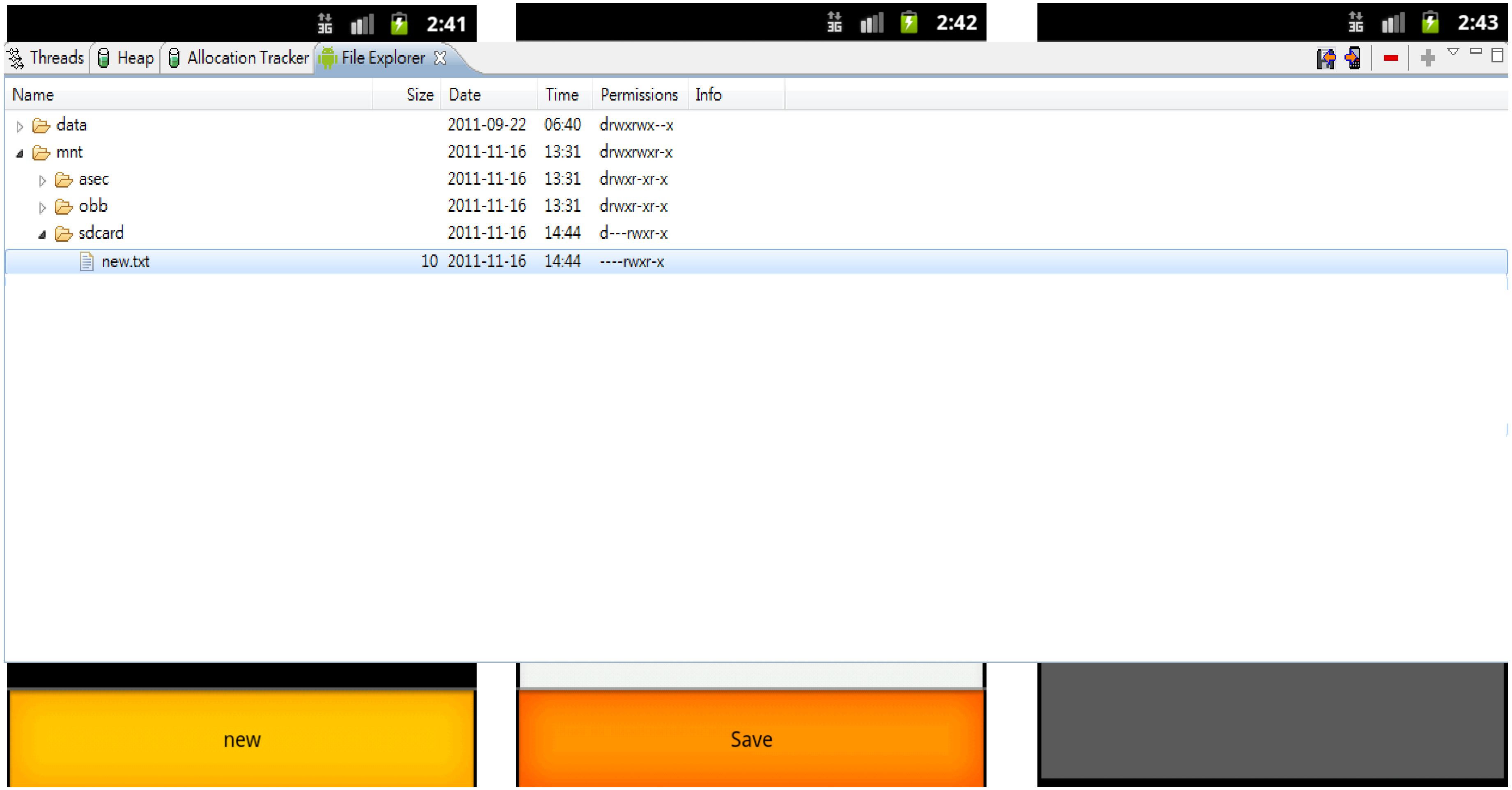
- Sample link:
  - http://pllab.cs.nthu.edu.tw/~cllee/Course/AndroidLab6_1.zip
- Try to create a new text file and store in sdcard
- Three errors of this sample

Dialog

```
LayoutInflater factory = LayoutInflater.from(this);
final View textEntryView = factory.inflate(R.layout.save_dialog, null);
Builder mBuilder1 = new AlertDialog.Builder(AndroidLab6_newFile.this);
mBuilder1.setView(textEntryView);

// Get 'EditText' from the dialog
myDialogEditText = (EditText) textEntryView
.findViewById(R.id.myDialogEditText);
myDialogEditText.setText(fileName);
```

- This class is used to instantiate layout XML file into its corresponding View objects
- *Inflate* a new view hierarchy from the specified xml resource
- Reference
  - http://developer.android.com/reference/android/view/LayoutInflater.html

Sample view



Name	Size	Date	Time	Permissions	Info
data		2011-09-22	06:40	drwxrwx--x	
mnt		2011-11-16	13:31	drwxrwxr-x	
asec		2011-11-16	13:31	drwxr-xr-x	
obb		2011-11-16	13:31	drwxr-xr-x	
sdcard		2011-11-16	14:44	d---rwxr-x	
new.txt	10	2011-11-16	14:44	----rwxr-x	

ListActivity & Adapter

What is List Activity and adapeter

- ListActivity
  - An activity that **displays a list of items** by binding to a data source such as an array or Cursor, and exposes event handlers when the user selects an item
- Adapter
  - An Adapter object acts as a bridge between an **AdapterView** and the **underlying data for that view**.
  - The Adapter provides access to the data items
- ListAdapter
  - Extended Adapter that is the bridge between a ListView and the data that backs the list
    - ArrayAdapter
    - SimpleAdapter

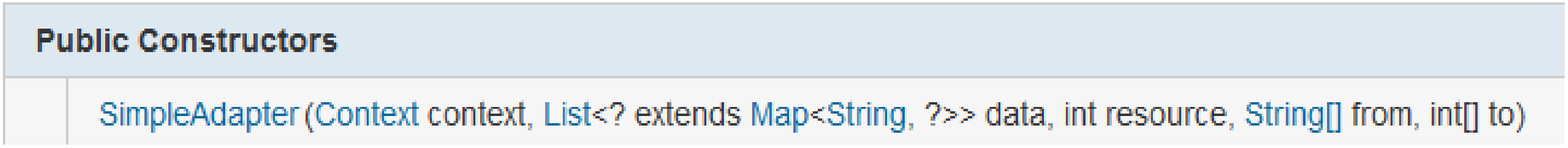
ArrayAdapter

- By default this class expects that the provided resource id references a single TextView
- However the TextView is referenced, it will be filled with the toString() of each object in the array
- You can add lists or arrays of custom objects

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.setListAdapter(this.createArrayAdapter());
    }
    private ArrayAdapter<String> createArrayAdapter() {
        String[] array = new String[] {
                "One", "Two", "Three", "Four", "Five"
        };
        return new ArrayAdapter<String>(this, R.layout.sample1, array);
    }
```

SimpleAdapter

- You can specify the data backing the list as an **ArrayList of Maps**
- Each entry in the ArrayList corresponds to one row in the list. **The Maps contain the data for each row**



Public Constructors

SimpleAdapter(Context context, List<? extends Map<String, ?>> data, int resource, String[] from, int[] to)

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.setListAdapter(this.createSimpleAdapter());}
```

```
private SimpleAdapter createSimpleAdapter() {
        List<Map<String, String>> data = this.createData();
        return new SimpleAdapter
        (this, data, R.layout.sample2,
                new String[] {"txt1", "txt2" },
                new int[] { R.id.txt1, R.id.txt2 });
}
```

SimpleAdapter

```java
private List<Map<String, String>> createData() {
        List<Map<String, String>> data = new ArrayList<Map<String, String>>();

        data.add(this.createMap("One", "Monday"));
        data.add(this.createMap("Two", "Tuesday"));
        data.add(this.createMap("Three", "Wednesday"));
        data.add(this.createMap("Four", "Thursday"));
        data.add(this.createMap("Five", "Friday"));

        return data;
}

private Map<String, String> createMap(String a, String b) {
        Map<String, String> map = new HashMap<String, String>();

        map.put("txt1", a);
        map.put("txt2", b);

        return map;
}
```

ListView

- A view that **shows items** in a vertically scrolling list. The items come from the **ListAdapter** associated with this view
- Event Listeners
  - setOnItemClickListner()
    - Register a callback to be invoked when an item in this AdapterView has been clicked
  - setOnItemLongClickListener()
    - Register a callback to be invoked when an item in this AdapterView has been clicked and held
  - setOnItemSelectedListener()
    - Register a callback to be invoked when an item in this AdapterView has been selected.

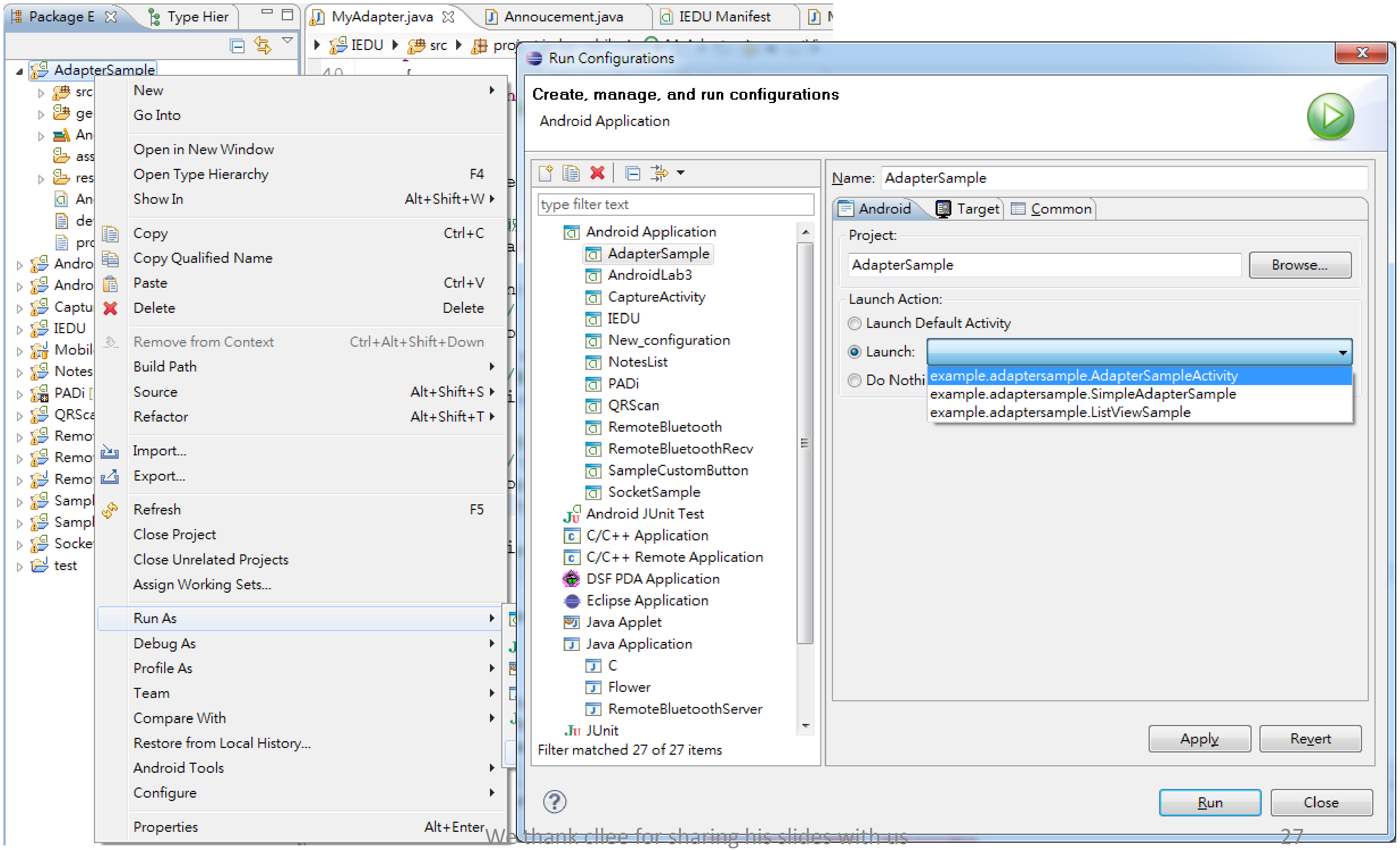
ListView

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        String[] mNames = new String[] { "One", "Two", "Three", "Four",
        "Five" };

        ArrayAdapter<String> datalist = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, mNames);

        setListAdapter(datalist);
        ListView lv = (ListView) this.findViewById(android.R.id.list);
        lv.setOnItemClickListener(this);
}

public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        Log.i("TAG", "onItemSelected: " + position);
}
```

Your time

- Sample link:
  - http://pllab.cs.nthu.edu.tw/~cllee/course/AndroidLab6_2.zip
- Different adapter samples
- Reference
  - http://developer.android.com/reference/android/widget/Adapter.html
- You can try to create your specific adapter

Choose different activities



Create, manage, and run configurations

Android Application

Name: AdapterSample

Project:

AdapterSample

Launch Action:

Launch Default Activity

Launch:

Do Nothi

example.adaptersample.AdapterSampleActivity

example.adaptersample.SimpleAdapterSample

example.adaptersample.ListViewSample

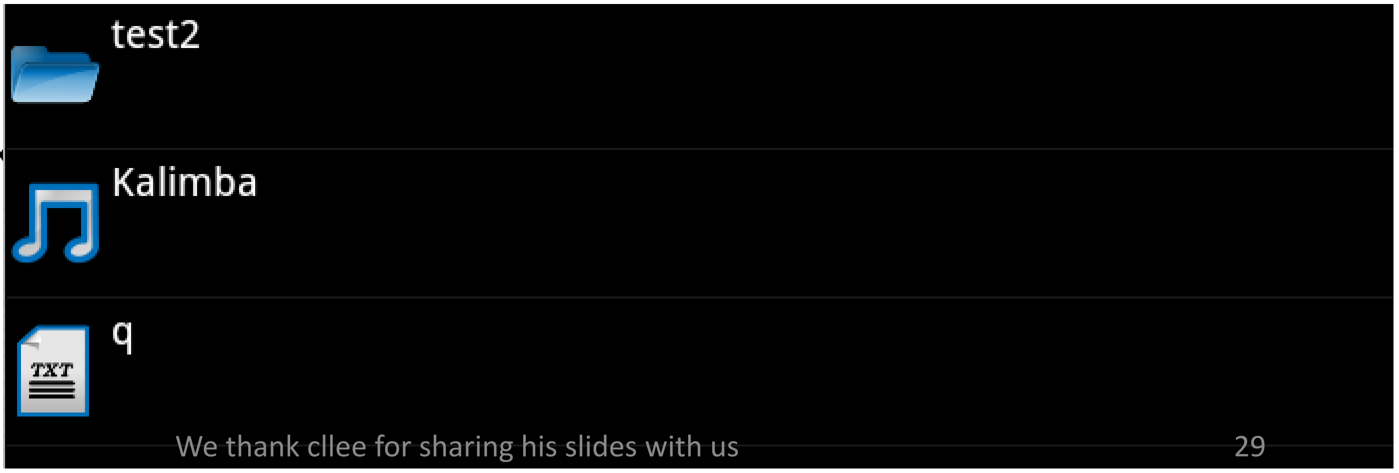
Filter matched 27 of 27 items

Lab Requirement

- Write a File Browser program
  - B+
    - Show the directories and files of sdcard
    - Create and delete text files
  - A-
    - Can update your file browser after create new file
  - A
    - Can open and show the created text file
  - A+
    - Different file type with different imageview
      - Text file, mp3 file, folder

Lab Requirement

- Write a File Browser program
  - B+
    - Show the directories and files of sdcard
    - Create and delete text files
  - A-
    - Can update your file browser after create new file
  - A
    - Can open and show the created text file
  - A+
    - Differe
      - Tex


test2

Kalimba

q