Permissions-demo
see PermissionDemo project on course website
NOTE: If your java has undefined symbols try hovering over the class, hit alt-enter and then import the class

In build.grade (app)
Define target SDK >23 (I choose 29) support down to 15 or so

Ensure the following dependencies exists in `build.gradle (app) version number is not important`

```
dependencies {
implementation 'androidx.appcompat:appcompat:1.3.1'
implementation 'com.google.android.material:material:1.4.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.0

}
```

In manifest ask for permission (goes above the application tag)
You may want to see if they are runtime (dangerous) permissions by searching for the exact permission (ie **`android.permission.CAMERA) at`**
https://developer.android.com/reference/android/Manifest.permission
we will only use the camera though

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

in mainactivity.xml
add a start camera button and an onClick listener (doCamera) for it
alt enter to create doCamera in main

```
<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:layout_marginBottom="8dp"
    android:text="Start Camera"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.54"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.151"
    android:onClick="doStartCamera"/>
```

in MainActivity.java

---

digression, show how you can add a click handler to the button using Java and an anonymous listener and how much extra scafolding you must put in verses the xml defined onClick handler.

```
Button b; //member var

...onCreate(){
b=findViewById(R.id.button);
b.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
    }
});

}
```

Back to permissions

add a couple of member vars to the Activity

```
//all the permissions we need (although we are only using the camera)
//this demonstrates how to ask for a bunch of permissions at one time
private static final String[]
PERMISSIONS={Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE,Manifest.permission.CAMERA};
private static final int PERMS_REQ_CODE = 200;
```

(BTW I'm cheating here, I'm going to use an intent to ask for the camera, that app will run under that apps permissions)

algorithm

click button 'StartCamera'

it checks permissions

if you have em then starts the camera

else

request camera permissions

```
public void startCamera(View view) {
    //TODO verify that app has permission to use camera
    //do we have needed permissions? if not
    if (!verifyPermissions())
        return;
    startCamera();
}
public void startCamera() {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivity(intent);
}
/**
 * Verify that the specific list of permisions requested have been granted,
otherwise ask for
 * these permissions.  Note this is coarse in that I assumme I need them all
 */
private boolean verifyPermissions() {
    //loop through all permissions seeing if they are ALL granted
    //iff ALL granted then return true
    boolean allGranted = true;
    for (String permission:PERMISSIONS){
        //a single false causes allGranted to be false
        allGranted = allGranted && (ActivityCompat.checkSelfPermission(this,
permission ) == PackageManager.PERMISSION_GRANTED);
    }
    if (!allGranted) {
        //OH NO!, missing some permissions, offer rationale if needed
        for (String permission : PERMISSIONS) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
permission)) {
                Snackbar.make(findViewById(android.R.id.content),
                        permission+" WE GOTTA HAVE IT!",
Snackbar.LENGTH_LONG).show();
            }
        }
        //Okay now finally ask for them
        requestPermissions(PERMISSIONS, PERMS_REQ_CODE);
    }
    //return whether they are granted or not
    return allGranted;
}
/***
 * callback from requestPermissions
 * @param permsRequestCode  user defined code passed to requestpermissions used to
identify what callback is coming in
 * @param permissions       list of permissions requested
 * @param grantResults      //results of those requests
 */
@Override
public void onRequestPermissionsResult(int permsRequestCode, String[] permissions,
int[] grantResults) {
    super.onRequestPermissionsResult(permsRequestCode, permissions, grantResults);
    boolean allGranted = true;
    switch (permsRequestCode) {
        case PERMS_REQ_CODE:
```

```
            for (int result: grantResults){
                allGranted = allGranted&&(result==
PackageManager.PERMISSION_GRANTED);
            }
            break;
    }
    if (allGranted)
        //TODO do your work here
        startCamera();
}
```