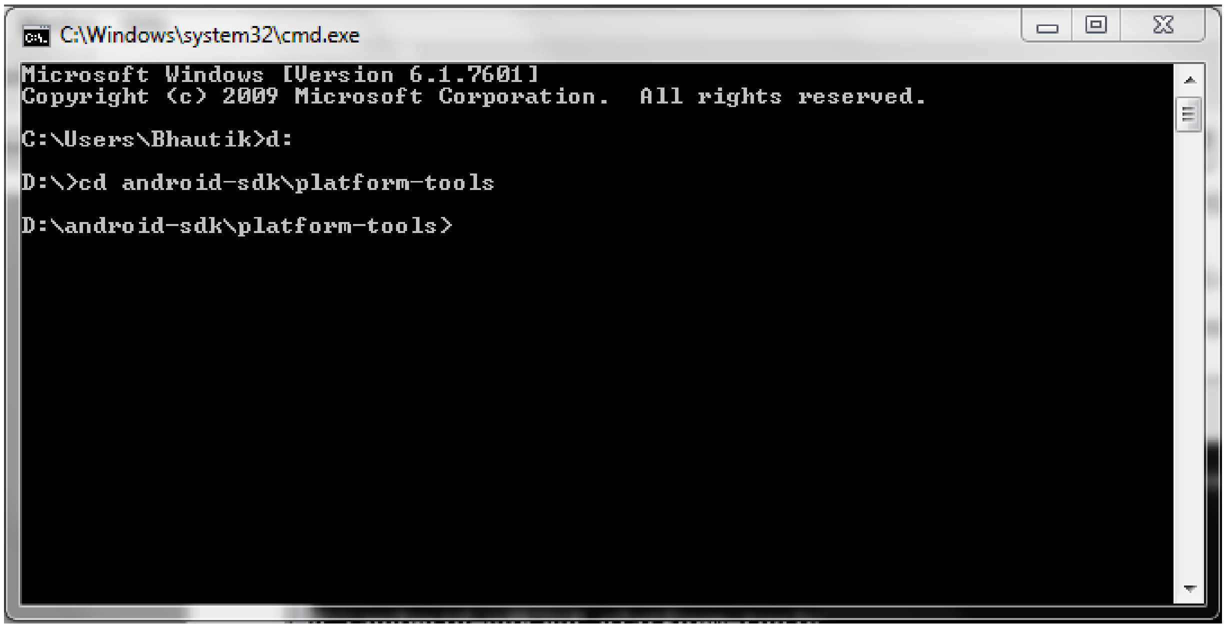
Step 1: First download sensorsimulator-xxx.zip from
https://code.google.com/archive/p/openintents/downloads?page=3

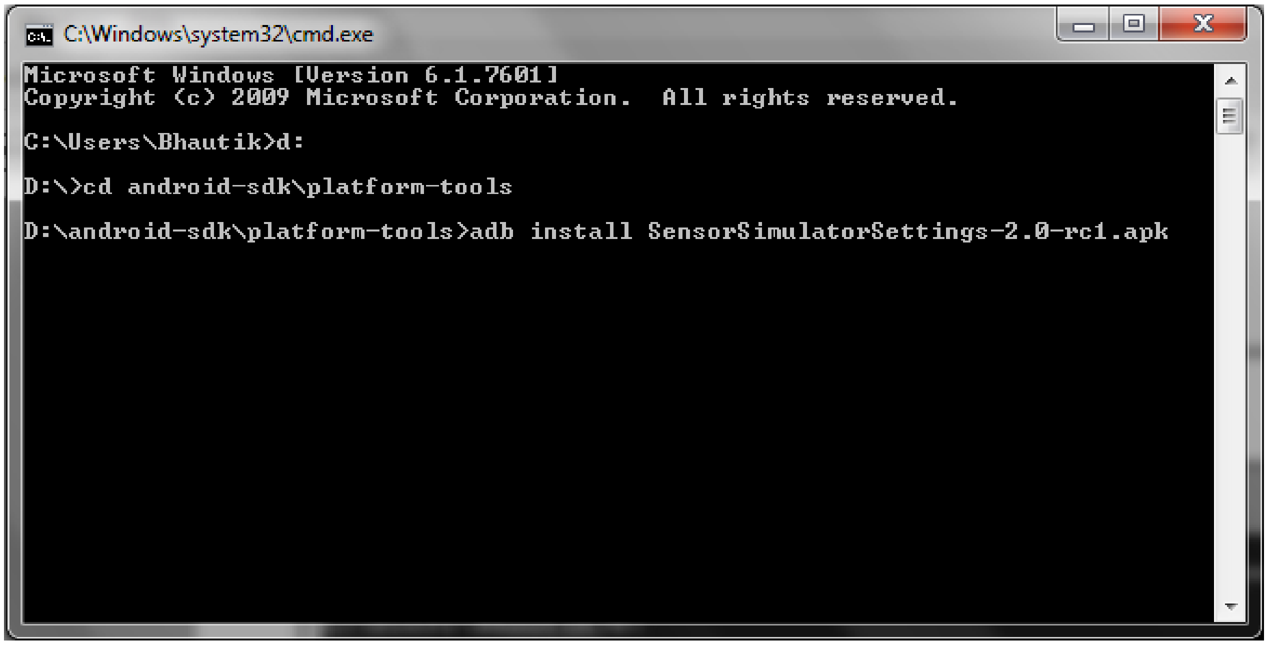
Step 2: Unzip the downloaded file and open sensorsimulator-xxx.jar file.

Step 3: Then copy sensorsimulatorsetting-xxx.apk to %ANDROID-SDK-HOME%/platform-tools folder, open your command prompt and redirect to folder where android sdk is installed. Then open platform-tools folder. Then write,



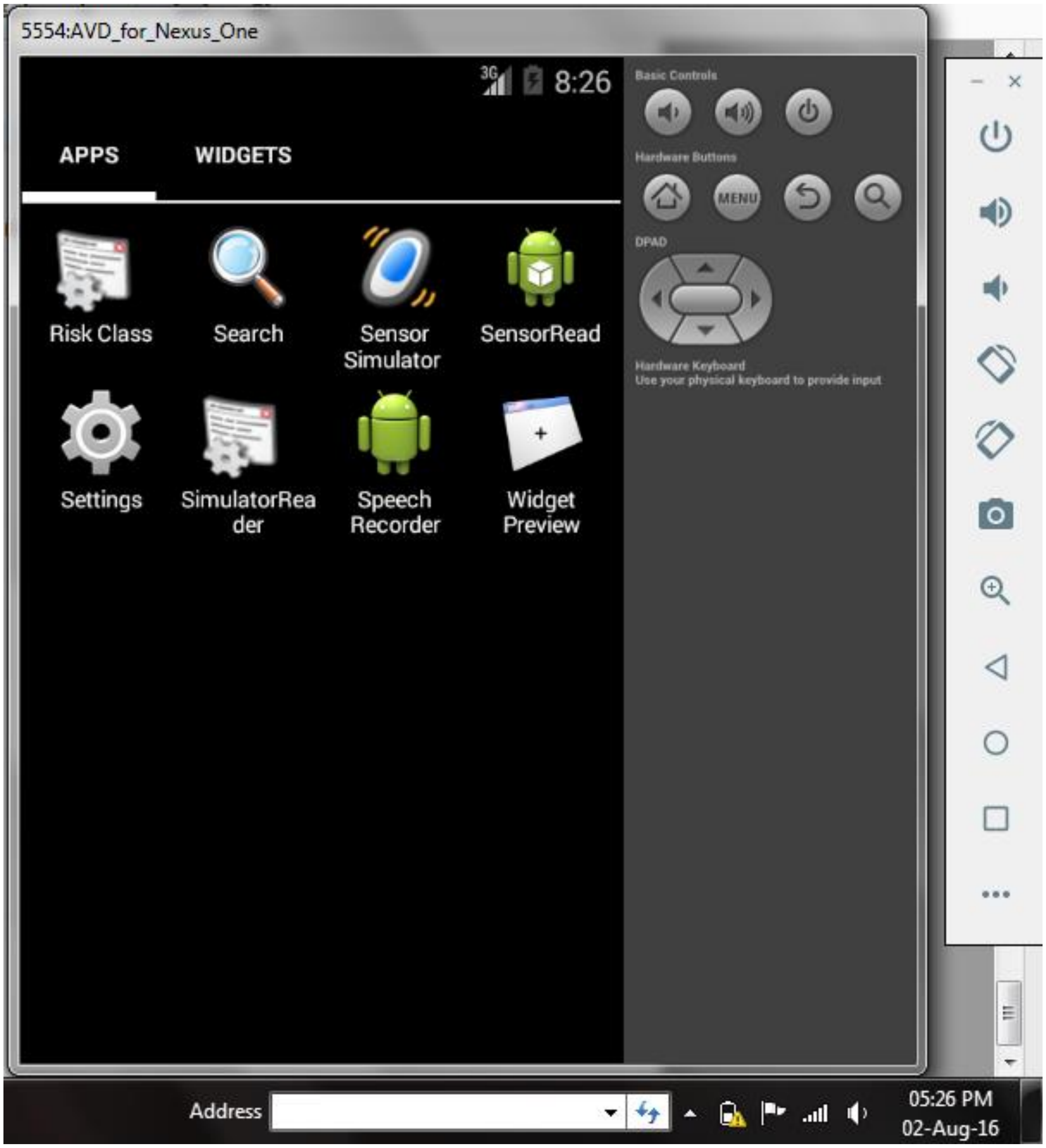
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Bhautik>d:

D:\>cd android-sdk\platform-tools

D:\android-sdk\platform-tools>
```

Then write bellow command to install sensorsimulatorsetting-xxx.apk.
"adb install sensorsimulatorsetting-xxx.apk".

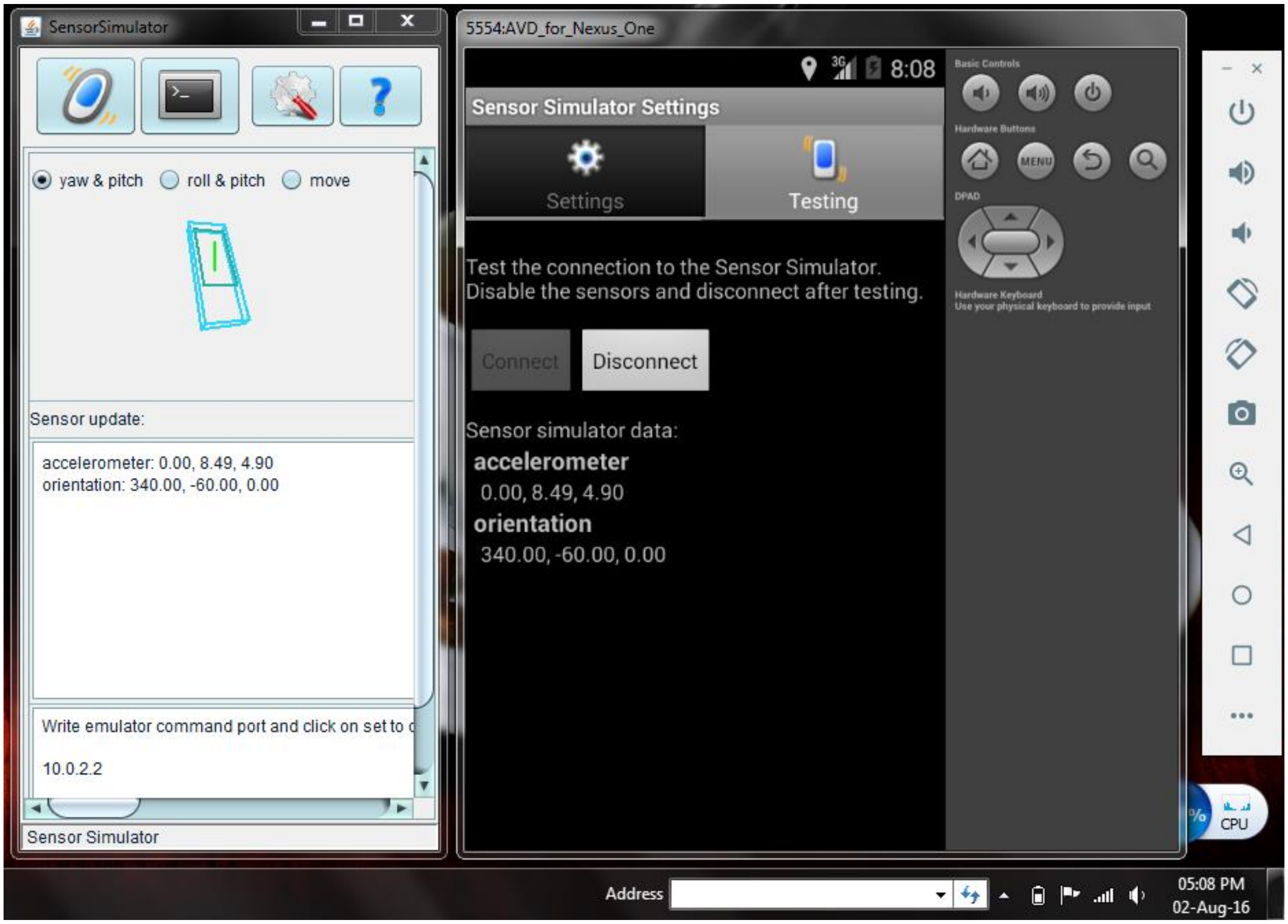


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Bhautik>d:

D:\>cd android-sdk\platform-tools

D:\android-sdk\platform-tools>adb install SensorSimulatorSettings-2.0-rc1.apk
```

Step 4: Now open your android virtual device and find and open sensor simulator app.



Step 5: Now check ip address provided in sensor simulator app and sensorsimulator.jar and click connect to connect with simulator. After that press home button in D-Pad and minimize this app.



```
accelerometer: 0.00, 8.49, 4.90
orientation: 340.00, -60.00, 0.00
```

Write emulator command port and click on set to c

10.0.2.2

Test the connection to the Sensor Simulator.
Disable the sensors and disconnect after testing.

Sensor simulator data:

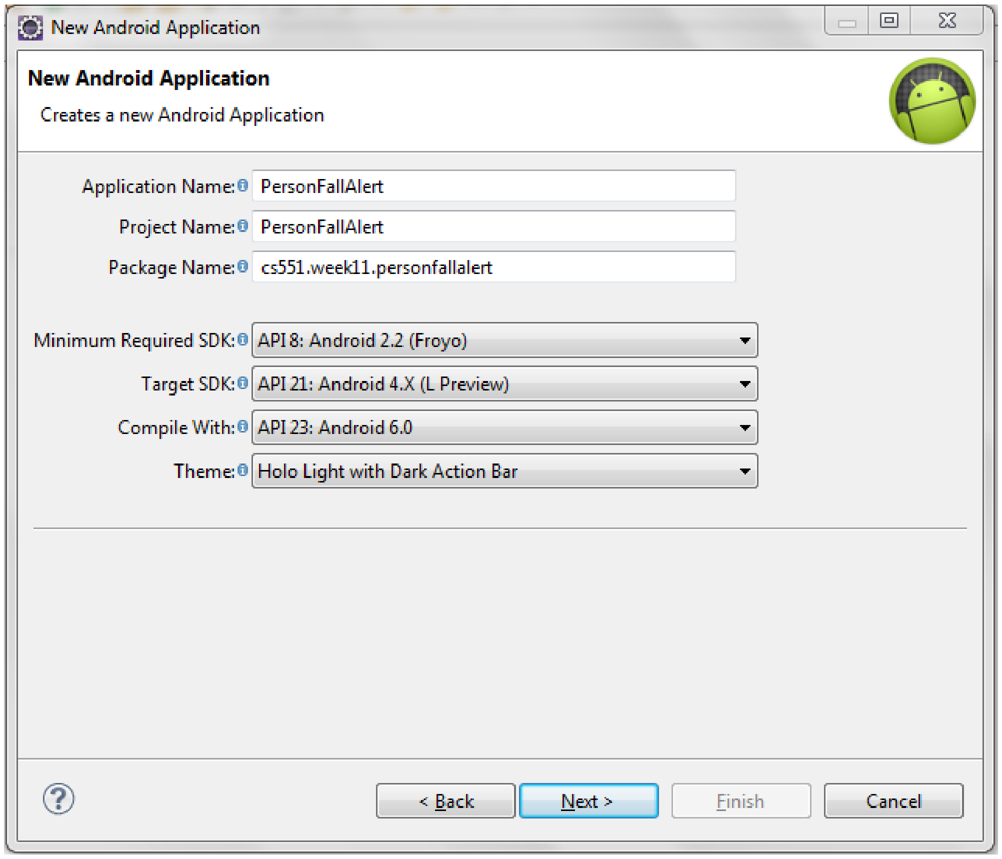
accelerometer

0.00, 8.49, 4.90

orientation

340.00, -60.00, 0.00

Step 6: Then create one android project and write code according to given sample in sensorsimulator-xxx.zip.


New Android Application

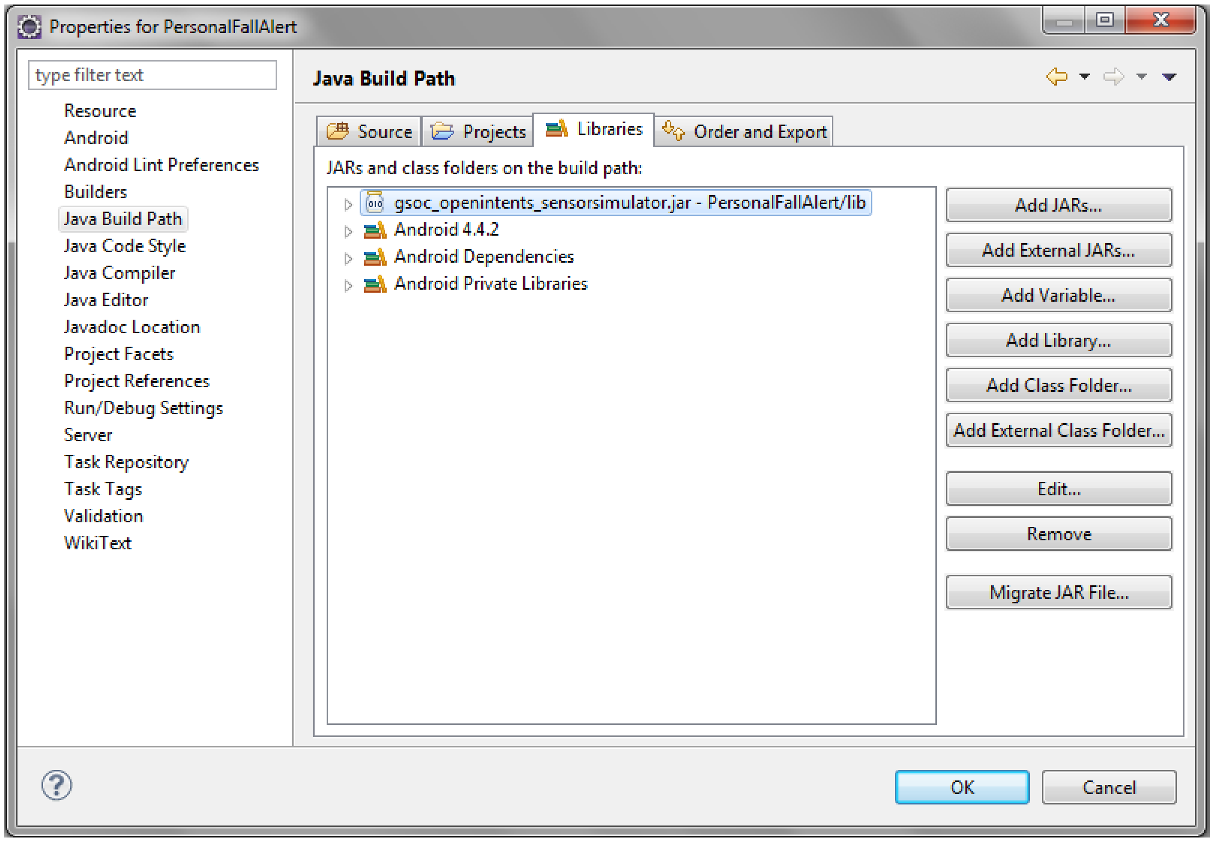
New Android Application

Creates a new Android Application

Application Name: PersonFallAlert
Project Name: PersonFallAlert
Package Name: cs551.week11.personfallalert

Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: API 21: Android 4.X (L Preview)
Compile With: API 23: Android 6.0
Theme: Holo Light with Dark Action Bar

Step 7: Now to connect with simulator and read its data. Import jar library provided in sensorsimulator/lib folder.


Properties for PersonalFallAlert

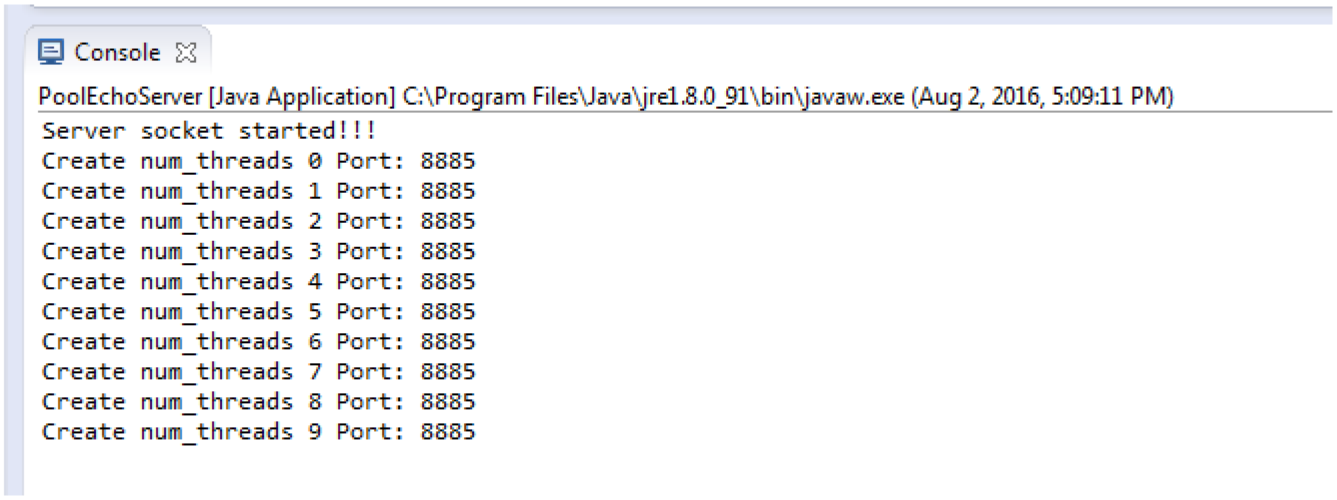
Java Build Path

JARs and class folders on the build path:

- gsoc_openintents_sensorsimulator.jar - PersonalFallAlert/lib
- Android 4.4.2
- Android Dependencies
- Android Private Libraries

Step 8: Now import sensorsimulatorsetting classes to your android project.

Step 9: Start Socket Server.

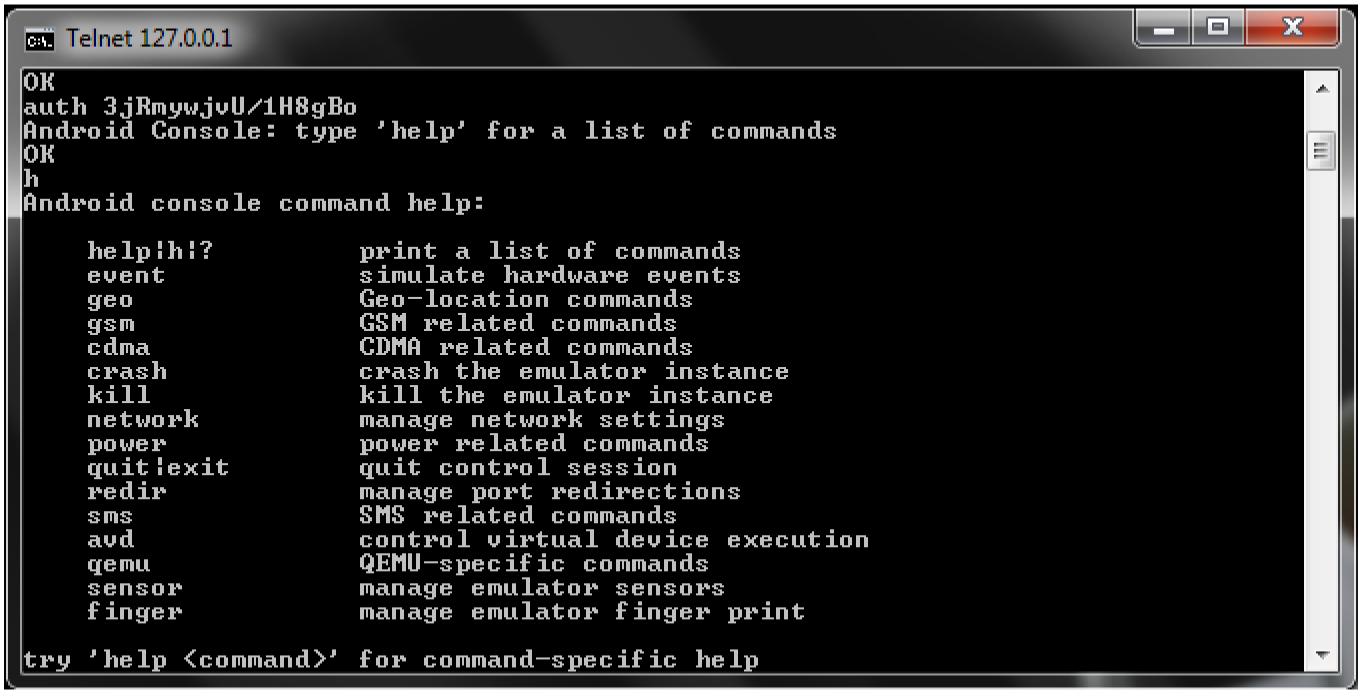


```
PoolEchoServer [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (Aug 2, 2016, 5:09:11 PM)
Server socket started!!!
Create num_threads 0 Port: 8885
Create num_threads 1 Port: 8885
Create num_threads 2 Port: 8885
Create num_threads 3 Port: 8885
Create num_threads 4 Port: 8885
Create num_threads 5 Port: 8885
Create num_threads 6 Port: 8885
Create num_threads 7 Port: 8885
Create num_threads 8 Port: 8885
Create num_threads 9 Port: 8885
```

Step 10: Now run android app into AVD. And after that click “Connect” button to connect with Socket Server.

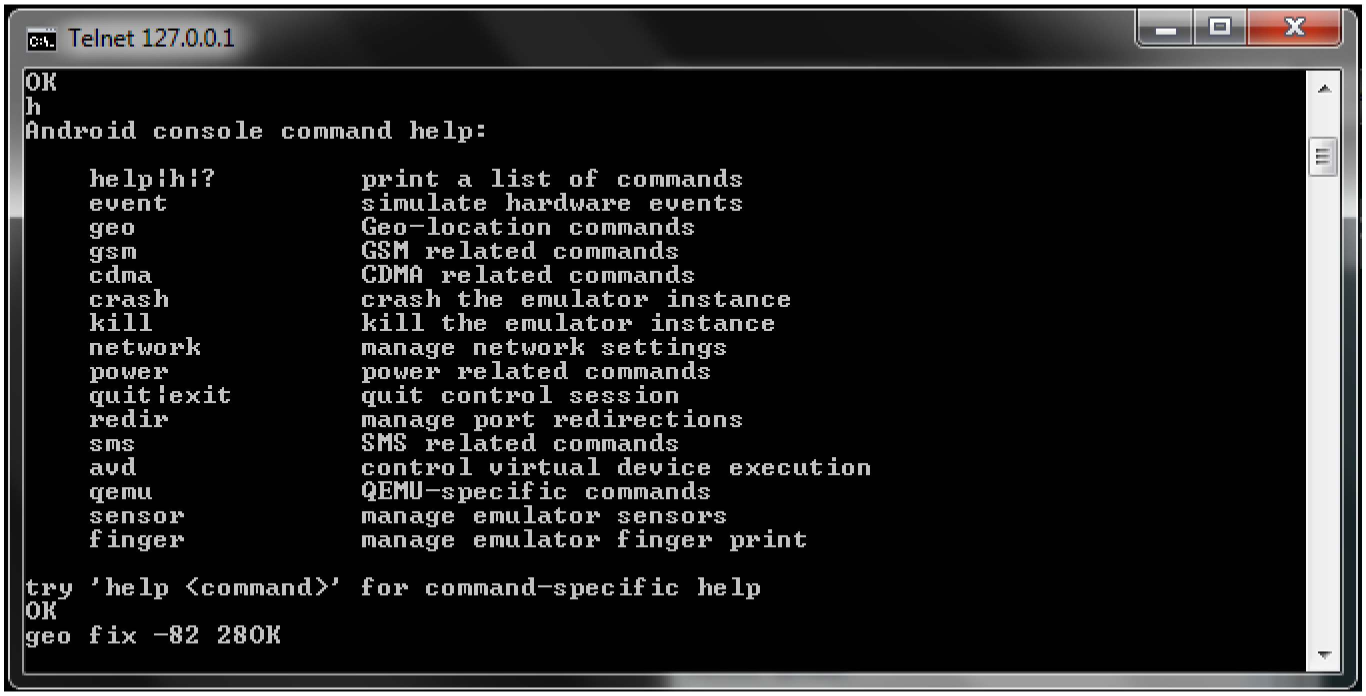
Step 11: You can see that there is not latitude and longitude shown in android app. For that connect with AVD using console with command. If you run android project in real device then it will take latitude and longitude using GPS.

“telnet 127.0.0.1 5554”



```
OK
auth 3jRmywjvU/1H8gBo
Android Console: type 'help' for a list of commands
OK
h
Android console command help:

    help|h|?         print a list of commands
    event            simulate hardware events
    geo              Geo-location commands
    gsm              GSM related commands
    cdma             CDMA related commands
    crash            crash the emulator instance
    kill             kill the emulator instance
    network          manage network settings
    power            power related commands
    quit|exit        quit control session
    redir            manage port redirections
    sms              SMS related commands
    avd              control virtual device execution
    qemu             QEMU-specific commands
    sensor           manage emulator sensors
    finger           manage emulator finger print

try 'help <command>' for command-specific help
```

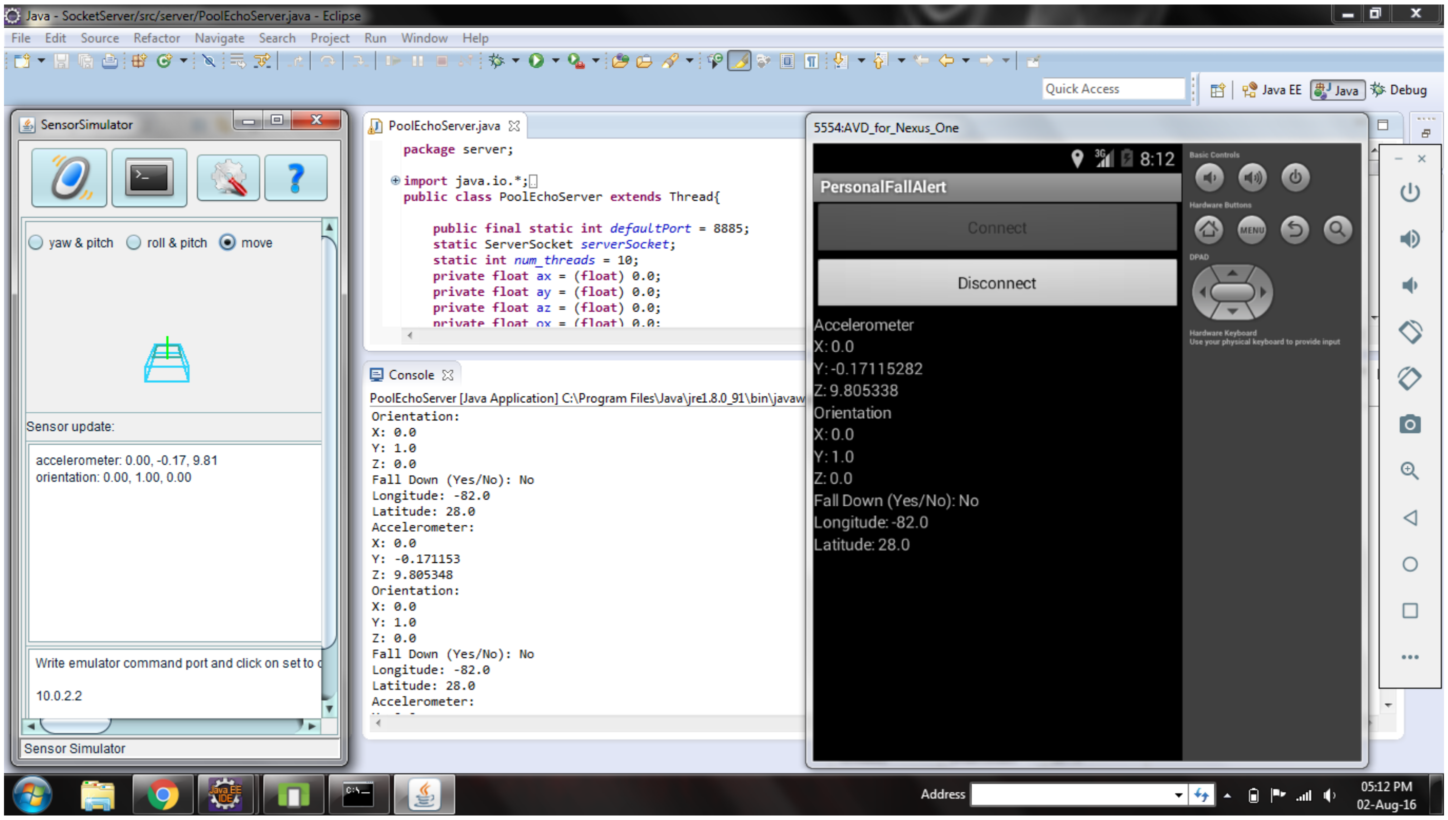
Now type “geo fix <latitude> <longitude>”



```
OK
h
Android console command help:

    help|h|?         print a list of commands
    event            simulate hardware events
    geo              Geo-location commands
    gsm              GSM related commands
    cdma             CDMA related commands
    crash            crash the emulator instance
    kill             kill the emulator instance
    network          manage network settings
    power            power related commands
    quit|exit        quit control session
    redir            manage port redirections
    sms              SMS related commands
    avd              control virtual device execution
    qemu             QEMU-specific commands
    sensor           manage emulator sensors
    finger           manage emulator finger print

try 'help <command>' for command-specific help
OK
geo fix -82 28OK
```

This command will set latitude and longitude.

Step 12: Now as your app is already connected to simulator using Sensor Simulator Setting App, now connecting with Server Socket, helps server socket to connect with simulator. Now any change in simulator will results into Server Socket. On Server Socket side, we will use Person Fall Alert to determine whether person is fall down or not and also shows latitude and longitude. If you run android project in real device then it will take latitude and longitude using GPS.



yaw & pitch [✓ move]

Sensor update:

accelerometer: 0.00, -0.17, 9.81
orientation: 0.00, 1.00, 0.00

Write emulator command port and click on set to c

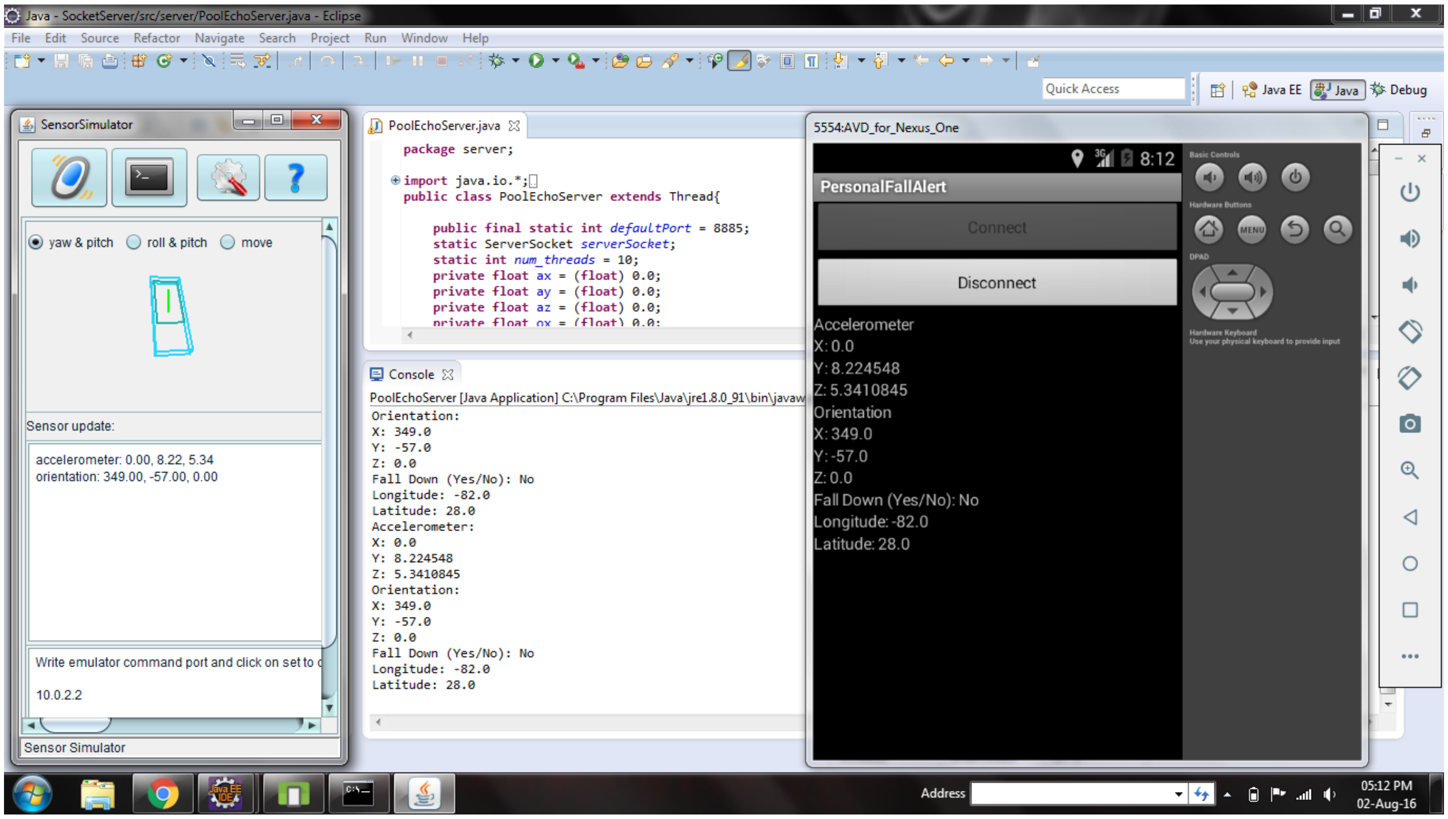
10.0.2.2

```
package server;

import java.io.*;
public class PoolEchoServer extends Thread{

    public final static int defaultPort = 8885;
    static ServerSocket serverSocket;
    static int num_threads = 10;
    private float ax = (float) 0.0;
    private float ay = (float) 0.0;
    private float az = (float) 0.0;
    private float ox = (float) 0.0;
```

PoolEchoServer [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw

```
Orientation:
X: 0.0
Y: 1.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
Accelerometer:
X: 0.0
Y: -0.171153
Z: 9.805348
Orientation:
X: 0.0
Y: 1.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
Accelerometer:
```

```
Accelerometer
X: 0.0
Y: -0.17115282
Z: 9.805338
Orientation
X: 0.0
Y: 1.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
```



[✓ yaw & pitch] move

Sensor update:

accelerometer: 0.00, 8.22, 5.34
orientation: 349.00, -57.00, 0.00

Write emulator command port and click on set to c

10.0.2.2

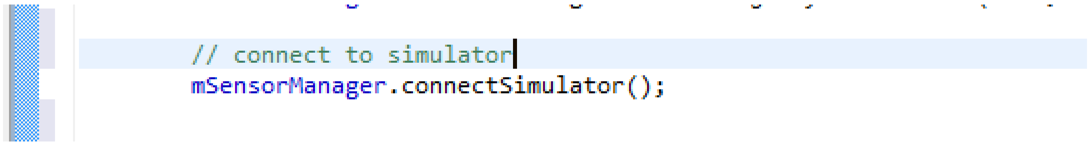
```
package server;

import java.io.*;
public class PoolEchoServer extends Thread{

    public final static int defaultPort = 8885;
    static ServerSocket serverSocket;
    static int num_threads = 10;
    private float ax = (float) 0.0;
    private float ay = (float) 0.0;
    private float az = (float) 0.0;
    private float ox = (float) 0.0;
```

PoolEchoServer [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw

```
Orientation:
X: 349.0
Y: -57.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
Accelerometer:
X: 0.0
Y: 8.224548
Z: 5.3410845
Orientation:
X: 349.0
Y: -57.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
```

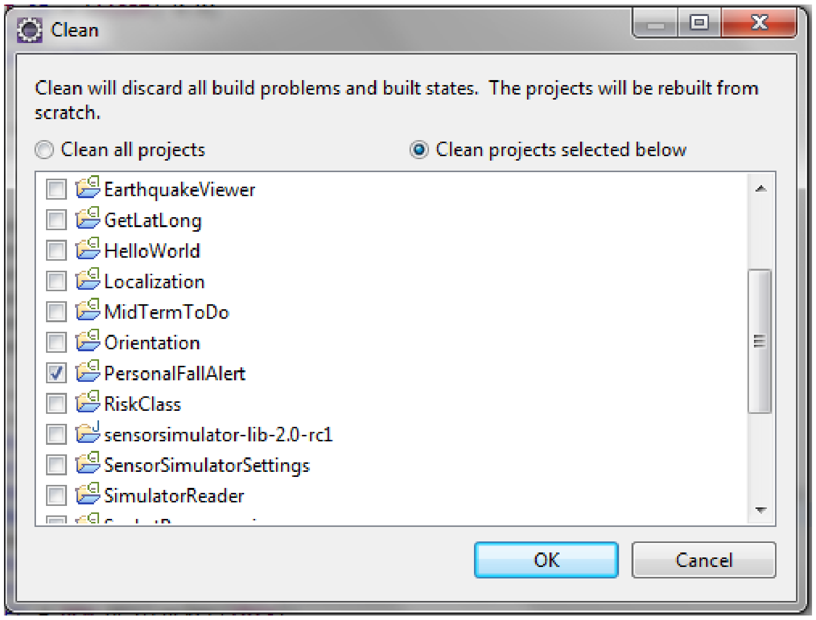
```
Accelerometer
X: 0.0
Y: 8.224548
Z: 5.3410845
Orientation
X: 349.0
Y: -57.0
Z: 0.0
Fall Down (Yes/No): No
Longitude: -82.0
Latitude: 28.0
```

Note: If you find an error like "java.lang.runtimeexception android.os.networkonmainthreadexception". Then create any class inside your activity class which extends AsyncTask class and make a connection with simulator inside doInBackgroun() method.



```
        // connect to simulator
        mSensorManager.connectSimulator();
```

And if you fine an error like "java.lang.classnotfoundexception" then try to clean your project. Sometimes, this may helps.



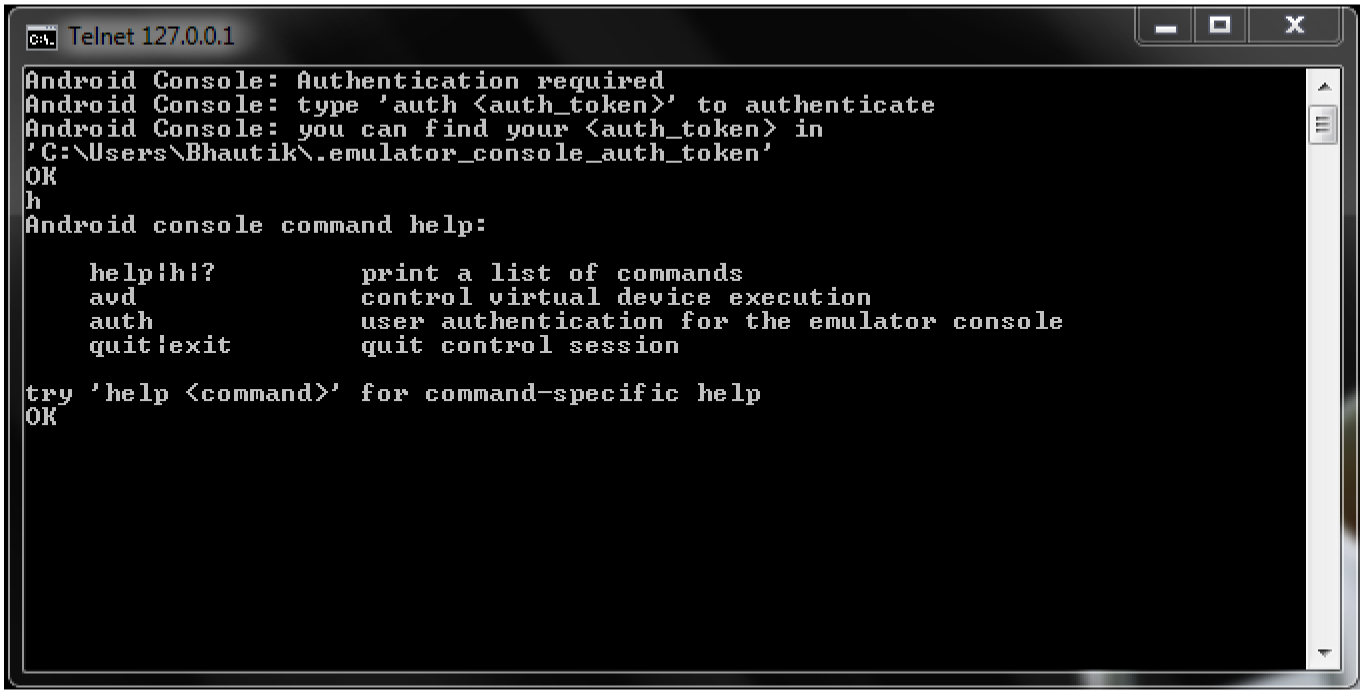
Clean

Clean will discard all build problems and built states. The projects will be rebuilt from scratch.

( ) Clean all projects [✓ Clean projects selected below]

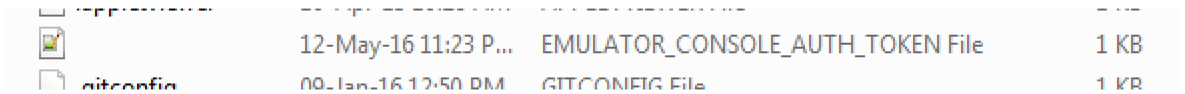
- [ ] EarthquakeViewer
- [ ] GetLatLong
- [ ] HelloWorld
- [ ] Localization
- [ ] MidTermToDo
- [ ] Orientation
- [x] PersonalFallAlert
- [ ] RiskClass
- [ ] sensorsimulator-lib-2.0-rc1
- [ ] SensorSimulatorSettings
- [ ] SimulatorReader

If you connect with AVD using console and type help and you only get command like bellow and could not find other commands.



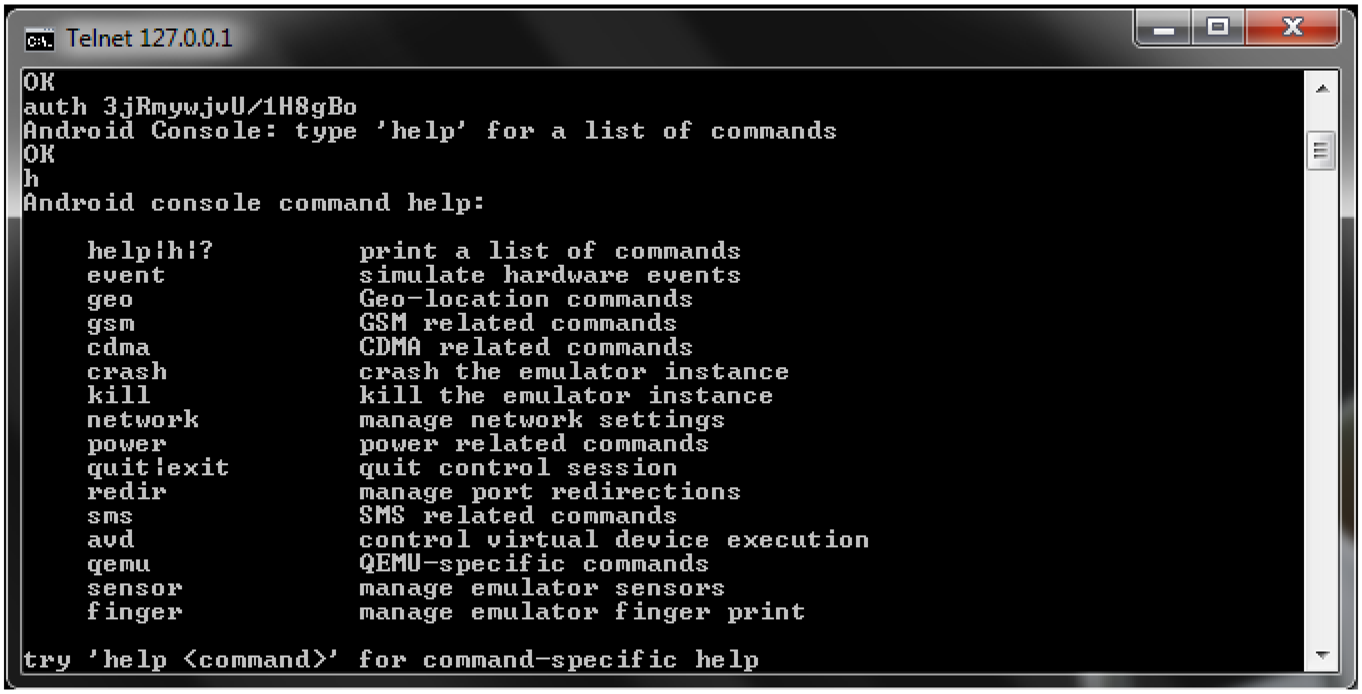
```
Telnet 127.0.0.1
Android Console: Authentication required
Android Console: type 'auth <auth_token>' to authenticate
Android Console: you can find your <auth_token> in
'C:\Users\Bhautik\.emulator_console_auth_token'
OK
h
Android console command help:

    help|h|?         print a list of commands
    avd              control virtual device execution
    auth             user authentication for the emulator console
    quit|exit        quit control session

try 'help <command>' for command-specific help
OK
```

Then go to you computer name folder, like my computer name is “Bhautik-PC”, then go to “C:\Users\Bhautik\”, then fine file “EMULATOR_CONSOLE_AUTH_TOKEN” and open it.



```
12-May-16 11:23 P...   EMULATOR_CONSOLE_AUTH_TOKEN File   1 KB
```

Now, copy the token and in cmd, type “auth <token_key>”.



```
OK
auth 3jRmywjvU/1H8gBo
Android Console: type 'help' for a list of commands
OK
h
Android console command help:

    help|h|?         print a list of commands
    event            simulate hardware events
    geo              Geo-location commands
    gsm              GSM related commands
    cdma             CDMA related commands
    crash            crash the emulator instance
    kill             kill the emulator instance
    network          manage network settings
    power            power related commands
    quit|exit        quit control session
    redir            manage port redirections
    sms              SMS related commands
    avd              control virtual device execution
    qemu             QEMU-specific commands
    sensor           manage emulator sensors
    finger           manage emulator finger print

try 'help <command>' for command-specific help
```

Using this you will fully connected with AVD.