```
1   /**
2    * Brocku.java
3    * Brock Butler
4    * Connects to the Brock University registrars' website to obtain course
5    * information from the current timetable
6    * Created by James Grisdale on 2013-02-24
7    * * Copyright (c) 2013 Sea Addicts. All rights reserved.
8   **/
9
10  package edu.seaaddicts.brockbutler.coursemanager;
11
12  import java.io.BufferedReader;
13  import java.io.InputStreamReader;
14  import java.net.URI;
15  import java.util.ArrayList;
16  import org.apache.http.client.HttpClient;
17  import org.apache.http.client.methods.HttpGet;
18  import org.apache.http.impl.client.*;
19  import org.apache.http.HttpResponse;
20  import android.os.AsyncTask;
21  //Using AsyncTask to do operations off the main thread
22  public class Brocku extends AsyncTask<Void, Void, ArrayList<MasterCourse>> {
23
24    /* doInBackground - connects to Brock University's registrar's office website to
      gather
25     * information on courses being offered, then returns an arraylist of MasterCourse
26     * objects which hold the data for all offerings at Brock.
27     */
28    protected ArrayList<MasterCourse> doInBackground(Void... Void) {
29      String codes[] = new String[74];
30      BufferedReader in = null;
31      String info = new String();
32      String substring = new String();
33      MasterCourse course;
34      boolean done;
35      ArrayList<MasterCourse> courseList = new ArrayList<MasterCourse>();
36      courseList.ensureCapacity(8000);
37      //test = "working";
38      try{
39        HttpClient client = new DefaultHttpClient();
40        HttpGet request = new HttpGet();
41        URI BTimeTable = new URI(
          "http://www.brocku.ca/registrar/guides/returning/timetable/a_get_subj.php?subj=C
          OSC");
42        request.setURI(BTimeTable);
43        HttpResponse response = client.execute(request);
44        in = new BufferedReader(new InputStreamReader(response.getEntity().getContent
          ()));
45        for (int i=0; i<3; i++) in.readLine();
46        for (int i=0; i<74; i++){
47          info = in.readLine();
48          codes[i] = info.substring(19,23);
49        }//retrieving all subjects
50        in.close();
51        for (int h=0; h<codes.length ; h++){
52          done = false;
53          BTimeTable = new URI(
            "http://www.brocku.ca/registrar/guides/returning/timetable/a_get_subj.php?subj
            ="+codes[h]);
54          request.setURI(BTimeTable);
55          response = client.execute(request);
56          in = new BufferedReader(new InputStreamReader(response.getEntity().getContent
            ()));
57          for (int i=0; i<98; i++) {in.readLine();}
58          info = in.readLine();
59          //for each subjects get all course offering information
60          if (info.length()<50){
61            while(!done){
62              course = new MasterCourse();
63              substring = info.substring(24, info.length() - 5);
64              course.id = substring;
```

```
65              for (int i=0; i<2; i++) {in.readLine();}
66              info = in.readLine();
67              substring = info.substring(123, 127);
68              course.subj = substring;
69              substring = info.substring(128, 132);
70              course.code = substring;
71              in.readLine();
72              info = in.readLine();
73              substring = info.substring(26,info.length() - 7);
74              course.desc = substring;
75              for (int i=0; i<4; i++) {in.readLine();}
76              info = in.readLine();
77              substring = info.substring(24,26);
78              course.dur = substring;
79              info = in.readLine();
80              substring = info.substring(24,info.length() - 5);
81              course.type = substring;
82              info = in.readLine();
83              substring = info.substring(24,info.length() - 5);
84              course.sec = substring;
85              info = in.readLine();
86              substring = info.substring(24,30);
87              course.days = substring;
88              info = in.readLine();
89              substring = info.substring(24,info.length() - 5);
90              course.time = substring;
91              info = in.readLine();
92              substring = info.substring(24,info.length() - 5);
93              if (info.substring(24,26).equals("<a"))
94                substring = info.substring(94, info.length()-9);
95              course.location = substring;
96              info = in.readLine();
97              if (info.length()>16)
98                substring = info.substring(9,info.length() - 5);
99              else substring = " ";
100             course.instructor = substring;
101             in.readLine();
102             in.readLine();
103             info = in.readLine();
104             courseList.add(course);
105             if (info.length() <20){
106               info = in.readLine();
107               done = true;
108             }
109           }
110         }
111         in.close();
112       }
113     }
114     //if there's an error return the error information in a course object
115     catch (Exception e){
116     info = e.toString();
117     course = new MasterCourse();
118     course.id = info;
119     courseList.add(course);
120     }
121     return courseList;
122   }
123
124   //not used
125   protected void onPostExecute(MasterCourse course){
126     posttest(course);
127   }
128
129   //not used
130   public MasterCourse posttest(MasterCourse course){
131     return course;
132   }
133 }
134
```