

```
<!--NCSU.java vs. CofC.java-->
```

```
2,3c2
```

```
< * jadvisor/school/NCSU.java - SchoolAdapter for North Carolina State Univ.
< * Copyright (C) 2001-2002 Curtis Rawls
```

```
---
```

```
> * jadvisor/school/COFC.java - SchoolAdapter for College of Charleston.
```

```
18a18,19
```

```
> *
> * CofC.java Authored by Chris Cargile - 9/23/13
```

```
29a31
```

```
>
>
34c36
< * North Carolina State University.
```

```
---
```

```
> * College of Charleston.
```

```
36,37c38
```

```
< * @author Curtis Rawls
< * @version 0.4.6
```

```
---
```

```
> * @author Chris Cargile
```

```
39,41c40,42
```

```
< public class NCSU implements SchoolAdapter {
< private static String CLASSES_URL = "http://www2.acs.ncsu.edu/reg_records/tracs_bk/";
< private static String COURSES_URL = "http://www2.acs.ncsu.edu/reg_records/crs_cat/";
```

```
---
```

```
> public class CofC implements SchoolAdapter {
> private static String CLASSES_URL = "
http://catalogs.cofc.edu/undergraduate/computer-science-bs-major-requirements.htm";
> private static String COURSES_URL = "
http://catalogs.cofc.edu/undergraduate/computer-science-bs-major-requirements.htm";
```

```
43c44,47
```

```
< new String[] {"fall", "sprg", "sum1", "sum2"};
```

```
---
```

```
> new String[] {"2013 Fall" , "2013 Summer" , "2013 Spring" , "2012 Fall" ,
> "2012 Summer" , "2012 Spring" , "2011 Fall" , "2011 Summer" , "2011 Spring" ,
> "2010 Fall" , "2009 Fall"
> };// "https://ssb.cofc.edu:9710/prod/bwckschd.p_disp_dyn_sched";
```

```
48c52,53
```

```
< private List[][] _classes;
```

```
---
```

```
> private List[][] _classes; // all the classes for a course ##
> // ie: ["110"][StudentClass(.."BIOL110"..),StudentClass(.."CSCI110"..)]
```

```
83c88
```

```
< public NCSU () {
```

```
---
```

```
> public CofC () {
```

```
85,98c90
```

```
< { "AGI", "ACC", "ADN", "AEE", "AFS", "ALS", "ANS", "ANT", "ARC",
< "ARE", "AS", "BAE", "BCH", "BIO", "BIT", "BMA", "BO", "BUS",
< "CBS", "CE", "CH", "CHE", "CNR", "COM", "COP", "CS", "CSC", "DDN",
< "DF", "E", "EAC", "EC", "ECG", "ECD", "ECE", "ECI", "ECO", "ED",
< "ELP", "EMS", "ENG", "ENT", "EOE", "ET", "FL", "FLC", "FLE", "FLF",
```

```

<         "FLG", "FLH", "FLI", "FLJ", "FLN", "FLP", "FLR", "FLS", "GRK", "LAT",
<         "FOR", "FPS", "FS", "FSA", "FW", "GC", "GD", "GEO", "GN", "GR", "HA",
<         "HI", "HON", "HS", "HSS", "ID", "IE", "IMM", "IMS", "LAR", "LOG", "M",
<         "MA", "MAE", "MAT", "MB", "MDS", "MEA", "MIS", "MLS", "MS", "MUS", "NE",
<         "NR", "NS", "NTR", "OR", "PA", "DAN", "PE", "PEC", "PEF", "PEH", "PEO",
<         "PHI", "PHY", "PMS", "PO", "PP", "PRT", "PS", "PSY", "PY", "REL", "SOC",
<         "SSC", "ST", "SVM", "SW", "T", "TAM", "TC", "TE", "TED", "TMS", "TOX",
<         "TT", "TTM", "VMA", "WGS", "WPS", "ZO"},
<     {} //full names for each department go here
<
< ---
<
< >         {"CSCI", "MATH", "PHYS", "GEOL", "CHEM", }
103c95,100
<
< ---
< > /**
< > * as implemented in DefaultSchoolAdapter, returned value will be a list all the
< > * departments. More helpfully, the system could benefit from a function that returns
< > * the true values of courses for a given semester. Would this be implemented via:
< > * a lookup against an ArrayList [semesters][courses] object?
< > */
133c130
<         throw new IOException("NCSU: Could not find department \"" + pre + "\"");
<
< ---
< >         throw new IOException("COFC: Could not find department \"" + pre + "\"");
137c134,135
<         throw new IOException("NCSU: Could not find course \"" + pre + " " + number + "\"");
<
< ---
< >         throw new IOException("COFC: Could not find course \"" + pre + " " + number + "\"");
< > System.out.println(((StudentClass)(_classes[i][j]).get(0)).getCourse().getCourseNumber()+"
" + _classes[i][j]);
193c191,202
<     a.add("Search Courses not yet implemented for NCSU.");
<
< ---
< >     for (int i = 0; i < _classes.length; i++) {
< >         if(_classes[i]!=null)
< >             for(int j=0 ; j < _classes[i].length;j++){
< >                 List classline = _classes[i][j];
< >                 for(Object o:classline){
< >                     StudentClass c = (StudentClass)o;
< >                     String s = c.toString();
< >                     if(s.contains(searchString))
< >                         System.out.println(searchString +" matched: "+ c);
< >                 }
< >             }
< >         }
214c223
<     return "NCSU SchoolAdapter";
<
< ---
< >     return "CofC SchoolAdapter";
217c226
< private StudentClass parseClassData (String s1, String s2, String s3) {
<
< ---
< > private StudentClass parseClassData (String s1, String s2, String s3, String s6) {

```

219,243c228,251

```

<  tokens[0] = s1.substring(132, 137);          //0 - COURSE PRE
<  tokens[1] = s1.substring(137, 142);          //1 - COURSE NUMBER
<  tokens[2] = s1.substring(142, 145);          //2 - COURSE NAME
<  tokens[3] = s1.substring(154, 173);          //3 - SECTION
<  tokens[4] = s1.substring(209, 217);          //4 - CALL NUMBER
<  tokens[5] = s1.substring(251, 263);          //5 - OPEN/RESTRICTED
<  tokens[6] = s1.substring(293, 297);          //6 - CREDIT
<  tokens[7] = s1.substring(342, 347);          //7 - CLASS SIZE
<  tokens[8] = s1.substring(391, 396);          //8 - OPEN SEATS AVAILABLE
<
<  tokens[9] = s2.substring(28, 36);             //9 - DAYS
<  tokens[10] = s2.substring(68, 80);            //10 - TIME
<  tokens[11] = s2.substring(113, 130);          //11 - TEACHER
<  tokens[12] = s2.substring(230, 247);          //12 - BUILDING
<  tokens[13] = s2.substring(260, 268);          //13 - ROOM NUMBER
<  tokens[14] = s2.substring(312, 317);          //14 - RESTR SEATS AVAILABLE
<  tokens[15] = s2.substring(360, 365);          //15 - WAIT LIST AVAILABLE
<
<  if (s3.length() > 0)
<      tokens[16] = s3.substring(26, s3.length() - 3); //16 - RESTRICTIONS
<  else
<      tokens[16] = "";
<
<  for (int i = 0; i < tokens.length; i++)
<      tokens[i] = tokens[i].trim();
---
>  if(s2.contains("+"))
>      s2=s2.replace("+", "");
>  tokens[0] = s2;                                //0 - COURSE PRE
>  tokens[1] = s1 ;                                //1 - COURSE NUMBER
>
>  tokens[2] = s3;                                //2 - COURSE NAME
>  tokens[3] = "002";                             //3 - SECTION
>  tokens[4] = "216320";                           //4 - CALL NUMBER
>  tokens[5] = "OPEN";                             //5 - OPEN/RESTRICTED
>  tokens[6] = s6;                                //6 - CREDIT
>  tokens[7] = "20";                               //7 - CLASS SIZE
>  tokens[8] = "10";                               //8 - OPEN SEATS AVAILABLE
>
>  tokens[9] = "M" ;                               //9 - DAYS
>  tokens[10] = "0910-1000" ;                       //10 - TIME
>  String altTime1 = "1100-1200" ;                 //10 - TIME
>  String altTime2 = "1300-1400";
>
>  tokens[11] = "rudolph";                          //11 - TEACHER
>  tokens[12] = "harrelson";                        //12 - BUILDING
>  tokens[13] = "02132";                            //13 - ROOM NUMBER
>  tokens[14] = "0" ;                               //14 - RESTR SEATS AVAILABLE
>  tokens[15] = "N/A";                             //15 - WAIT LIST AVAILABLE
>  tokens[16] = " " ;                               //16- RESTRICTIONS

```

251,253c259,261

```

<  TimeOfDay[] times = parseTime(tokens[10]);

```

```

<
<  int credit;
---
>
>  TimeOfDay[] times = parseTime(tokens[10]);
>  int credit;
257,259c265
<      credit = (int)Double.parseDouble(tokens[6]);
<
<  String building = tokens[12].substring(0, tokens[12].indexOf("</a>"));
---
>      credit = (int)Double.parseDouble(tokens[6]+".0");
383,399c397,431
<      while ((line = reader.readLine()) != null
<          && !(line.length() >= 5 && line.substring(0, 5).equals("<PRE>"))) {
<          //System.err.println("DEBUG: got " + line.length() + " bytes");
<      }
<  } catch (Exception e) {
<      throw new IOException("Invalid cache file: " + f);
<  }
<
<  try {
<      while ((line = reader.readLine()) != null) {
<          int i = line.indexOf("<font size=3><a name = ");
<          if (i >= 0) {
<              line = line.substring(i);
<              line2 = reader.readLine();
<              line3 = reader.readLine();
<              c = parseClassData(line, line2, line3);
<              if (!courseList.contains(c.getCourse().getCourseNumber()))
---
>      while ((line = reader.readLine()) != null){
>          if(line.length()>=76 && line.contains("<p class=\"courses\"><a\"")){
>              line=line.substring(75,line.length());
>              dept=line.substring(0,4);
>              course=line.substring(5,9);
>              course=course.trim();
>              if(line.contains("href=")){
>                  int ctEnd=line.indexOf(" href=");
>                  courseTitle=line.substring(0,ctEnd-1);
>                  String prefix=dept + " " + course;
>                  courseTitle=courseTitle.replace(prefix+" ", "");
>                  credit=courseTitle.substring(courseTitle.length()-2,courseTitle.length()-1);
>                  courseTitle=courseTitle.substring(0,courseTitle.length()-4);
>                  System.out.println(prefix+","+courseTitle+","+credit);
>              }
>              c = parseClassData(course, dept, courseTitle, credit);
>              TimeOfDay[] times = parseTime("1100-1200");
>              TimeOfDay[] times2 = parseTime("1200-1300");
>              TimeOfDay[] times3 = parseTime("1400-1500");
>
>              TimeOfDay start1 = times[0];
>              TimeOfDay end1 = times[1];

```

```

>         TimeOfDay start2 = times2[0];
>         TimeOfDay end2 = times2[1];
>
>         TimeOfDay start3 = times3[0];
>         TimeOfDay end3 = times3[1];
>         StudentClass alt1=new StudentClass(c);
>         StudentClass alt2=new StudentClass(c);
>         StudentClass alt3=new StudentClass(c);
>         alt1.setTime(start1, end1);
>         alt2.setTime(start2, end2);
>         alt3.setTime(start3, end3);
>         //StudentClass c2 = alt1.getStudentClass();
>         if (!courseList.contains(c.getCourse().getCourseNumber())){
401c433
<             classList.add(c);
---
>         //         courseList.add(new String(c.getCourse().getCourseNumber()));}
402a435,439
>             classList.add(c);
>             classList.add(alt1);
>             classList.add(alt2);
>             classList.add(alt3);
>         }

<!-------Advisor.java----->
248c248
<         new SchoolAdapter[] {new NCSU(), new UNC(), new DefaultSchoolAdapter()};
---
>         new SchoolAdapter[] {new CofC(), new NCSU(), new UNC(), new DefaultSchoolAdapter()};

```