Coursesv7 Documentation

Authors: Declan Holmes, Jonathan Hein, Sean Ackerley

Purpose:

Provide an easy and efficient means to populate the database for the ATAR calculator project with courses listed on the VTAC website, without the need for manual insertion.

Libraries:

- Selenium Client & WebDriver
  - URL: http://selenium-release.storage.googleapis.com/2.45/selenium-java-2.45.0.zip
  - JAR's to be included in build path:
    - Selenium-java-2.45.0-srcs.jar
    - Selenium-java-2.45.0.jar
- Selenium Server
  - URL: http://selenium-release.storage.googleapis.com/2.45/selenium-server-standalone-2.45.0.jar
  - JAR's to be included in build path:
    - Selenium-server-standalone-2.45.0.jar
- Com.mysql.jdbc_5.1.5

Installation Environment:

- The generated JAR for the project is expected to reside and execute on the machine in which the target MySQL server resides.
- The project is designed without a GUI, that is to say, it was created for an environment in which textual console output is supported.

Description of operation:

UpdateCourses.java:

- Starting point of the program.
- Navigates to VTAC website, performs course search with zero search parameters to retrieve a list of all courses currently offered.
- Searches the html source code for a defined pattern by which it identifies a course.
  - Once the identifying pattern is found a substring containing just the course code is created and entered into the arrayList courseNum.
  - This process repeats for the length of the page.
- For each course code in CourseNum arrayList:
  - Creates a new thread with a GetInfoRunnable object as its parameter.
  - Starts the thread.

    - Main thread sleeps for 50 milliseconds so as not to bombard the web server with requests.
    - Main thread joins to all created threads to ensure it doesn’t continue until all threads have finished.
- Once all threads have finished, for each course in the courses arrayList uses the DBProcessor singleton object to enter the course into the database.

GetInfoRunnable class:

- Navigates to the URL given as a parameter.
- Performs the findName method
    - Searches the page source for anything that matches one of the name patterns defined in the class.
    - Returns the name of the course residing between the defined start pattern and defined end pattern.
- Performs the findUni method
    - Searches the page source for anything that matches the UNI name patterns defined in the class.
    - Returns the name of the institution residing between the defined start pattern and defined end pattern.
- Locates and processes the ‘clearly in’ ATAR if one exists.
- If course isn’t a graduate course, course is added to the courses arrayList supplied as a parameter.

Course class:

- Encapsulates course data.
- Provides getter methods for access.

DBProcessor class:

- Singleton class, only one instance is allowed to exist at a given time. Accesses by other classes by means of the getInstance method.
- On construction, creates a connection to the database described in the MYSQL_LOC variable.
- EnterCourse method:
    - Takes a course as a parameter.
    - Calls the enterUni method.
    - Executes an insert operation containing the information provided in the course parameter on the courses table.
- EnterUni method:
    - Takes a course a parameter
    - Queries the UNI table for entries where name is equal to the institution name provided in the course parameter.
    - If the result set return from the above query returns nothing, inserts a new entry in the UNI table for that institution.

- Returns the uid for the institution.

Setup/Maintenance:

Given that the program is reliant on the format of a resource outside of our realm of control, maintenance is required for continued functionality. Many of the patterns used to identify the data this program hopes to gather are likely to change at some point or another, this may necessitate looking at page source code to find identifying features of desired data and updating the relevant pattern accordingly. Below is a list of the variables which are likely to require maintenance as well as their description.

Variables	**Description**
UpdateCourses class	
- COURSE_LINK	Identifies a course link in the search results page source. To extract just the relevant course code, a start index and end index are specified which define the substring.
- VTAC	Defines the address for the search page. Also used in the concatenated string supplied to a GetInfoRunnable thread to define the course web page to extract data from.
- MID_YEAR	Defines the 'mid-year' string for use during mid-year in take. Must be removed during the 'yearly' intake.
- COURSE_EX	Concatenated with the VTAC variable as well as relevant course code, to be supplied to a GetInfoRunnable Thread.
GetInfoRunnable class	
- BACH_PATTERN	A temporary variable used only for a single course in the list which did not follow the standard format.
- MONTH_PATTERN	Used to identify the starting point of an ATAR table.
- HE_PAT	Identifies courses that are have the HE classification.
- VET_PAT	Identifies courses that are have the VET classification.
- GRAD_PAT	Identifies courses that are have the GRAD classification.
- HEGRAD_PAT	Identifies courses that are have the HEGRAD classification.
- HEVET_PAT	Identifies courses that are have the HEVET classification.
- HEGET_PAT	Identifies courses that are have the HEGET classification.
- INST_PAT	Identifies institution name.
DBProcessor	
- MYSQL_LOC	Defines the database URL as well as its user and password.

	May need to be redefined in the event the program is used on a database where that information would be incorrect.
- SQL_INSERT_COURSE	Defines the insert statement to be used to enter data into the course table. May need to be redefined in the event the structure of the database changes.
- SQL_INSERT_UNI	Defines the insert statement to be used to enter data into the uni table. May need to be redefined in the event the structure of the database changes.
- SQL_SELECT_ALL_UNI	Defines the select query for retrieving institutions from the uni table based on the name supplied by the relevant course. May need to be redefined in the event the structure of the database changes.

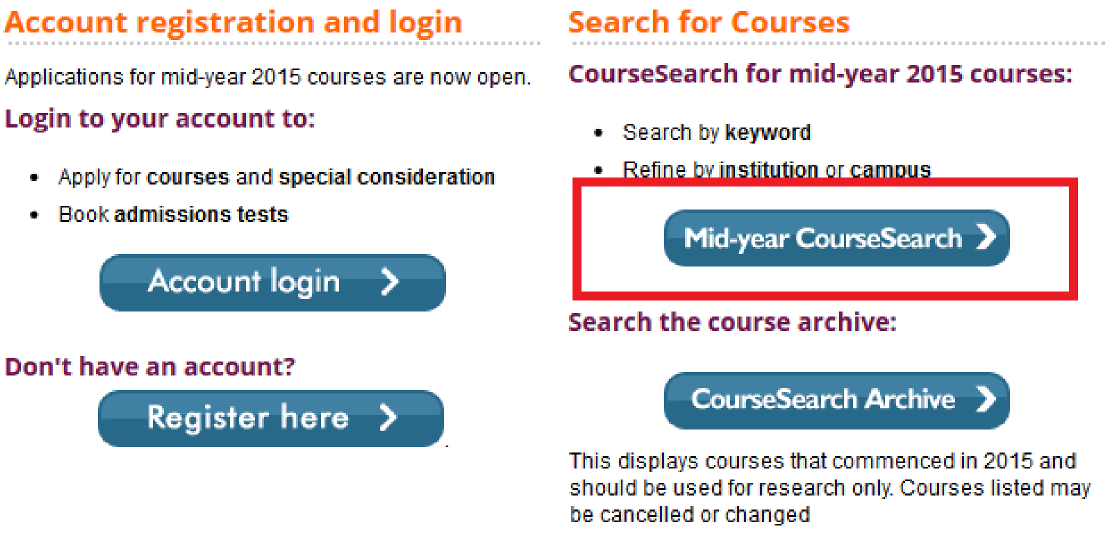
Switching between start of year, mid-year and archive modes:

- At the start of execution, the user is prompted for a single key input:
  - ‘s’ for start of year mode.
  - ‘m’ for mid-year mode.
  - ‘a’ for archive mode.
- Selecting the wrong mode for the corresponding time of year may result in the program working incorrectly or not at all.

Steps for ensuring program is navigating to the search results page:

1. Ensure the VTAC variable in UpdateCourses.java correctly corresponds to the URL found (figure 2) when clicking on the ‘CourseSearch’ button (figure 1) on the VTAC website minus any extension (such as ?midYear=1).

Figure 1.



Account registration and login

Applications for mid-year 2015 courses are now open.

Login to your account to:

- Apply for courses and special consideration
- Book admissions tests

Don't have an account?

Search for Courses

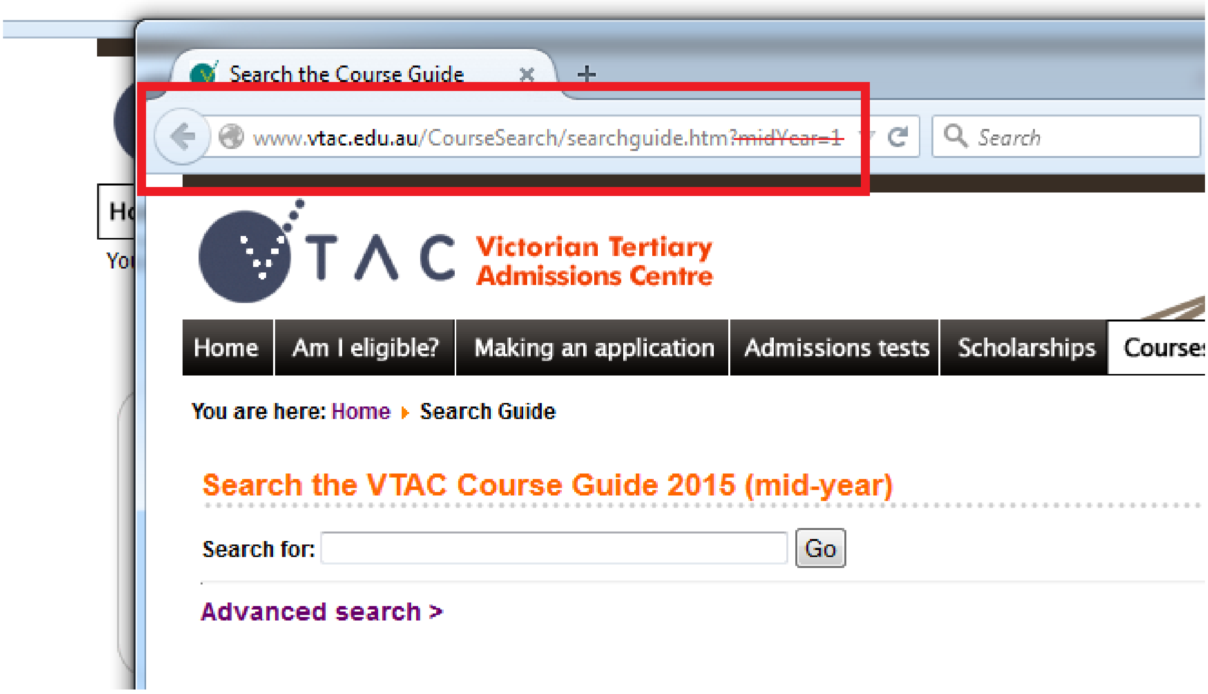
CourseSearch for mid-year 2015 courses:

- Search by keyword
- Refine by institution or campus

Search the course archive:

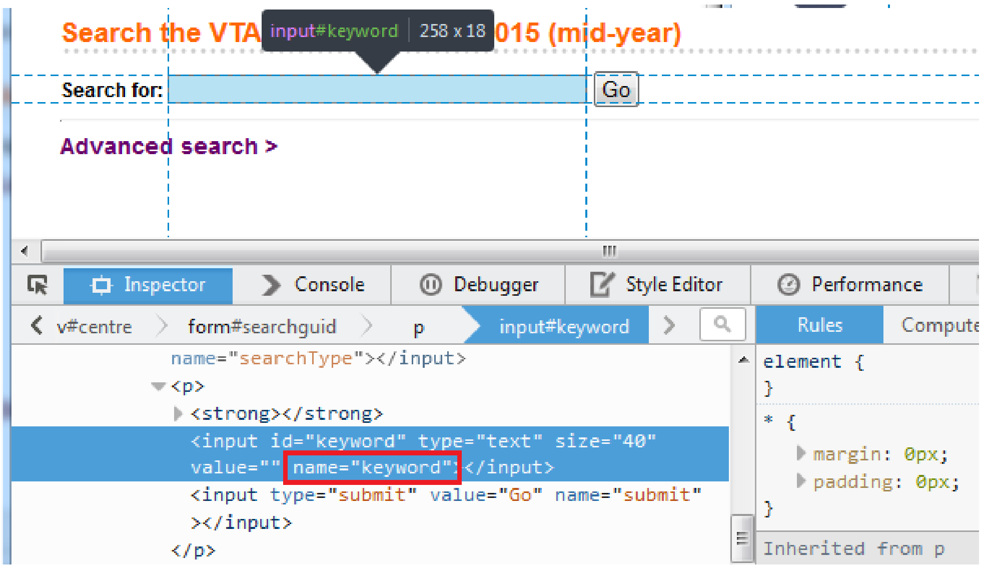
This displays courses that commenced in 2015 and should be used for research only. Courses listed may be cancelled or changed

Figure 2. Displaying what the VTAC variable should be.


www.vtac.edu.au/CourseSearch/searchguide.htm~~?midYear=1~~

2. If it is currently mid-year, the MID_YEAR variable in UpdateCourses.java may require updating. The substring of the URL above crossed out above should be used as the value of this variable (?midYear=1).
3. The ARCH variable in UpdateCourses.java may also require updating. The means of doing so is the same as the above but clicking on the ‘CourseSearch Archive’ button in place of the normal current courses button.
4. If after confirming the above is correct the program presents the error message ‘Search button element not found, ensure the SEARCH_BUTTON variable is set correctly by inspecting the <u>text field</u> in your browser (figure 3).

Figure 3.



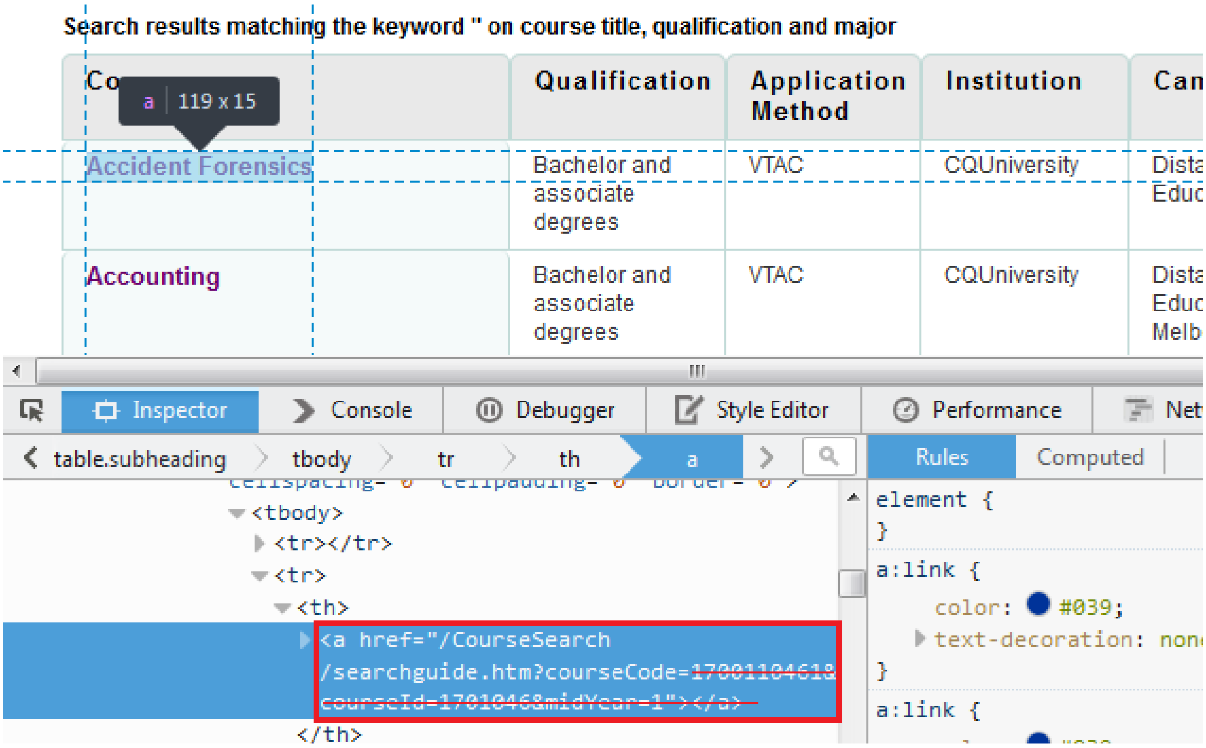
```
name="searchType"></input>
<p>
  <strong></strong>
  <input id="keyword" type="text" size="40"
  value="" name="keyword"></input>
  <input type="submit" value="Go" name="submit"
  ></input>
</p>
```

```
element {
}
* {
  margin: 0px;
  padding: 0px;
}
Inherited from p
```

Steps for ensuring the program will correctly identify courses listed from the output of the above:

1. Inspect any course element in the table returned by clicking the ‘Go’ button seen in figure 3. Set the COURSE_LINK variable in UpdateCourses.java to the identified link element (figure 4).

   Figure 4.



Search results matching the keyword '' on course title, qualification and major

Co	Qualification	Application Method	Institution	Can
Accident Forensics	Bachelor and associate degrees	VTAC	CQUniversity	Dista Educ
Accounting	Bachelor and associate degrees	VTAC	CQUniversity	Dista Educ Melb

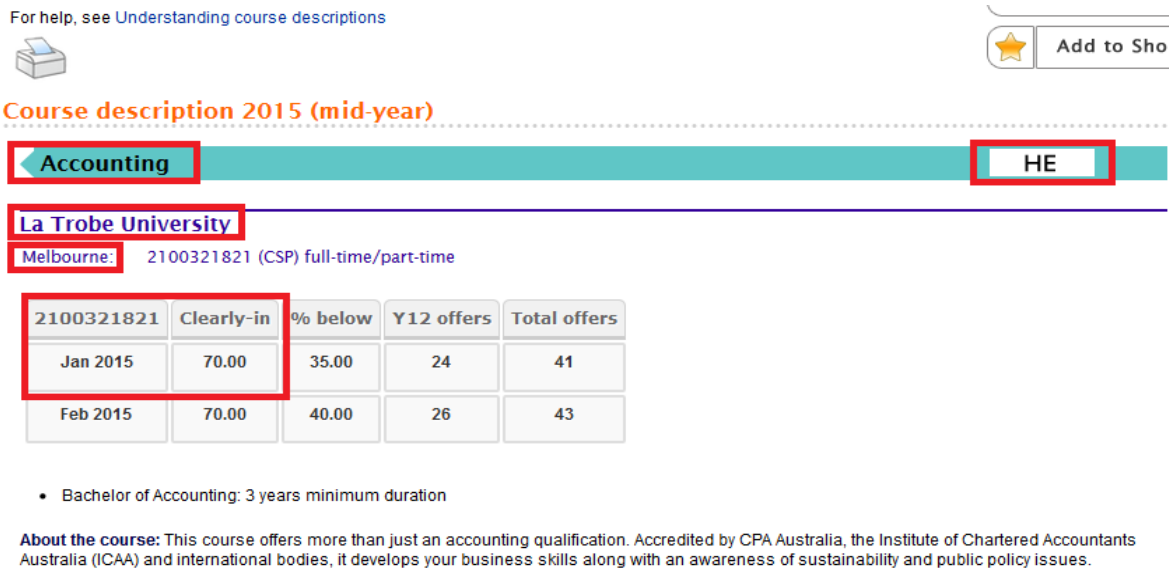
```
    cellspacing="0" cellpadding="0" border="0">
  <tbody>
    <tr></tr>
    <tr>
      <th>
        <a href="/CourseSearch
        /searchguide.htm?courseCode=~~17001104618~~
        ~~courseId=1701046&midYear=1"></a>~~
      </th>
```

```
element {
}
a:link {
    color: #039;
    text-decoration: none
}
a:link {
```

Ensuring ‘field identifier variables’ are correct in CoursesRunnable.java:

On any course page, there are a number of fields we are interested in (figure 5).

Figure 5.



For help, see Understanding course descriptions

Course description 2015 (mid-year)

Accounting HE

La Trobe University

Melbourne: 2100321821 (CSP) full-time/part-time

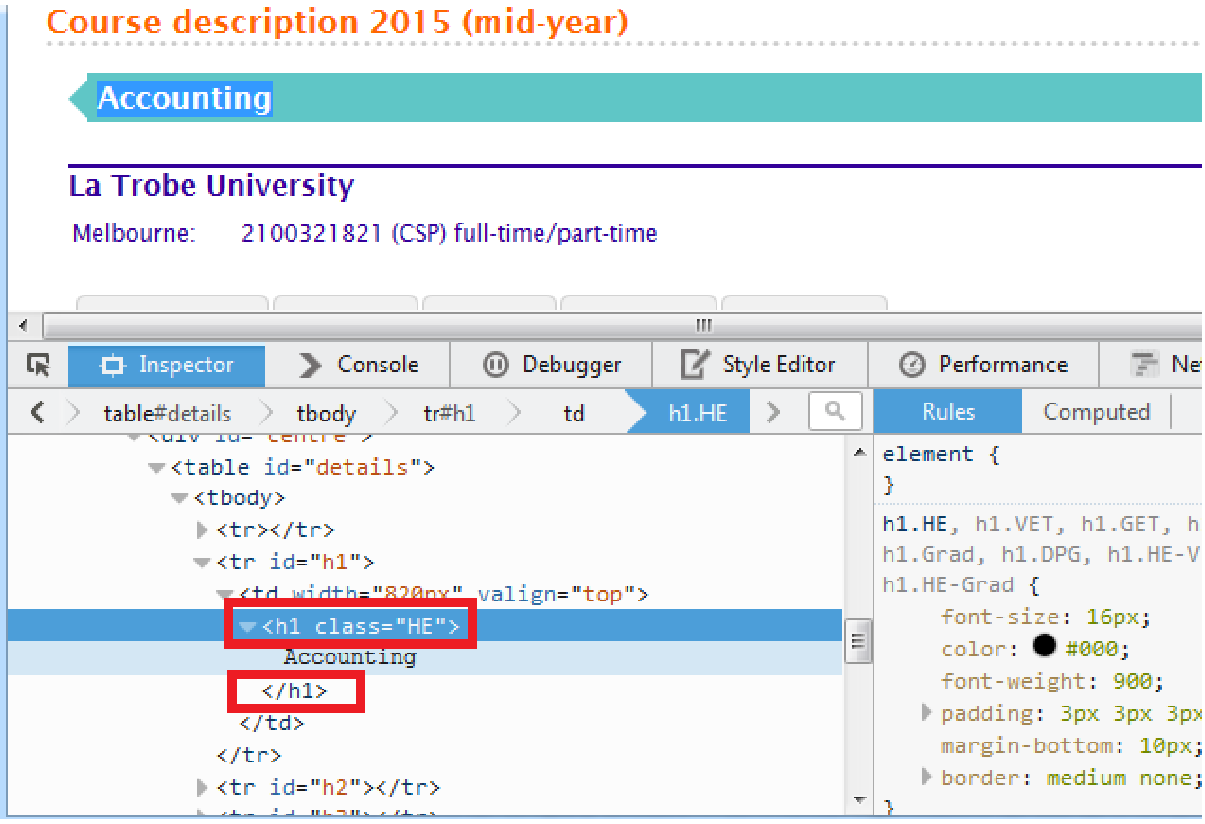
2100321821	Clearly-in	% below	Y12 offers	Total offers
Jan 2015	70.00	35.00	24	41
Feb 2015	70.00	40.00	26	43

- Bachelor of Accounting: 3 years minimum duration

About the course: This course offers more than just an accounting qualification. Accredited by CPA Australia, the Institute of Chartered Accountants Australia (ICAA) and international bodies, it develops your business skills along with an awareness of sustainability and public policy issues.

Each of these elements may be inspected to find its identifying features (figure 6).

Figure 6. Finding the enclosing features for identifying a desired field.



Course description 2015 (mid-year)

Accounting

La Trobe University

Melbourne: 2100321821 (CSP) full-time/part-time

```
      <div id="centre">
        <table id="details">
          <tbody>
            <tr></tr>
            <tr id="h1">
              <td width="820px" valign="top">
                <h1 class="HE">
                  Accounting
                </h1>
              </td>
            </tr>
            <tr id="h2"></tr>
            <tr id="h3"></tr>
```

```
element {
}
h1.HE, h1.VET, h1.GET, h
h1.Grad, h1.DPG, h1.HE-V
h1.HE-Grad {
    font-size: 16px;
    color: ● #000;
    font-weight: 900;
    padding: 3px 3px 3px
    margin-bottom: 10px;
    border: medium none;
}
```

The above example displays the identification of a course name. '<h1 class="HE">' identifies the start, and '</h1>' identifies the end of the content desired. These strings are placed in the variables "HE_PAT" and "H1_END" respectively.

The above process applies to all desired fields identified in the 'CoursesRunnable' variable table (page 3).