Teaching Tip
Using Cascading Style Sheets to Design a Fly-Out Menu with Microsoft Visual Studio

Chang Liu
Charles Downing
OM&IS Department
Northern Illinois University
DeKalb, IL 60115, USA
cliu@niu.edu, cdowning@niu.edu

ABSTRACT

The menu has become an integrated component within nearly all professionally designed websites. This teaching tip presents a no-code way to design either a vertical or a horizontal fly-out menu by using Cascading Style Sheets (CSS) within Microsoft Visual Studio 2008. The approach described in this tip helps students fully understand how to visually create a professional menu for a website, and also serves as an introduction to CSS.

Keywords: Fly-Out, Menu, Design, Web, CSS, Visual Studio, Expression Web

1. INTRODUCTION

Increasingly, people go online for communication, interaction, searching, and buying products or services. Recent statistics indicate the number of Internet users worldwide reached over 1.80 billion as of December 31, 2009 (Internet Usage Statistics, 2010). This represents a global penetration rate of 26.6% which means over one fourth of the total world population is composed of Internet and Web users. One of the critical challenges facing businesses today is to develop a website that is "usable", as usability has been shown to correlate with site usage. Two key components of usability are "Ease of Use" and "Emotion" ("affective reactions invoked by a Web site"), and thus companies continually seek more effective ways to make professional and appealing sites which are easy to navigate (Downing and Liu, 2010). Innovative menus are receiving much attention in this area.

A menu is a key part of any dynamic website that features more than one page. It allows users to navigate a website quickly and easily and allows them to get where they want to go. The menu tends to be the fundamental interaction component and contributes to the important aspect of "stickiness", or usage for a website design (Nielsen, 2000). A good menu design makes a huge difference in terms of whether or not users could effectively navigate and achieve a visually appealing effect of consistent look and feel when they come to visit a website (Baxley, 2003). Since the menu has become an integrated component within almost all professionally designed websites, and innovative menus mesh strongly with both the "Ease of Use" and "Emotion" aspects of site usability, the ability to create professional and innovative web menus is a highly prized skill for both

students and seasoned developers. This teaching tip describes a no-code way to design either a vertical or horizontal fly-out menu as one visually appealing example in the classroom. It allows students with little or no programming experience the ability to create a functional and appealing web menu. This is the motivation for the teaching tip which follows.

2. THE HANDS-ON ENVIRONMENT

The lesson described in this paper was based on a semester-long undergraduate level Web design course taught in the spring of 2009 at a large, Midwestern university. All students enrolled in this course were MIS majors. The lead author of this paper was the course instructor. One of the learning objectives in this course was requiring students to explore Cascading Style Sheets (CSS) to control the appearance of Web pages. CSS is a collection of rules of how a Web page should be formatted (Duckett, 2008). It is widely understood by all major browsers and overcomes the problem of mixed data and presentation in a Web page by allowing all formatting information, i.e., the rules, to be stored in a separate external CSS file. This helps create an easily accessible and maintainable website since the page content and the formatting rules are separated (Accessibility Features of CSS, 1999).

The problem of using HTML markup tags for formatting and presentation is that markup tags are embedded into the content which means that visual significance of the content could only be processed the way determined by the designer. Moreover, using HTML markup tags to specify formatting details is not a good long-term solution because documents become big, cumbersome, and

impossible to maintain. CSS, on the other hand, was designed to allow precise control – outside of markup – of character spacing, text alignment, object position on the page, audio and speech output, font characteristics, etc. By separating style from markup, a web designer can simplify and make web contents more accessible at the same time. In fact, accessibility is an important part of website usability since a website is not usable unless it's accessible (Krug, 2006). Students learned at the end of this course to use different rules of CSS to control the appearance of every element in their websites so that their Web design looked more accessible, appealing and interesting.

Microsoft Visual Studio 2008 (VS 2008) was used for the development platform in this class since it is designed to create interactive and dynamic HTML pages for a website. VS 2008 has sophisticated support for CSS design. It enables developers to visually create CSS, making it much easier to style Web pages without the need to remember the detailed code of CSS (Spaanijaars, 2008). Moreover, one of the big improvements of VS 2008 compared with previous versions of Microsoft Visual Studio tool is its dramatically improved HTML designer, and the rich CSS support it brings. It supports a new visual tool called "Manage Styles" for CSS. This allows visually designed Web page formats in CSS without writing a single line of code. In short, CSS aided by VS 2008 provides designers a friendly and easy way to access and control the appearance of Web pages.

3. THE FLY-OUT MENU DESIGN ASSIGNMENT

The instructor asked the students to use the CSS tool of VS 2008 to complete a menu design. In the class, the students were required to visit a particular website, Dell Computer (*http://www.dell.com*), to examine its menu component. Figure 1 shows the main menu from Dell's site. The purpose of the visit was to ensure that the students received a real-world assignment in the classroom learning environment.
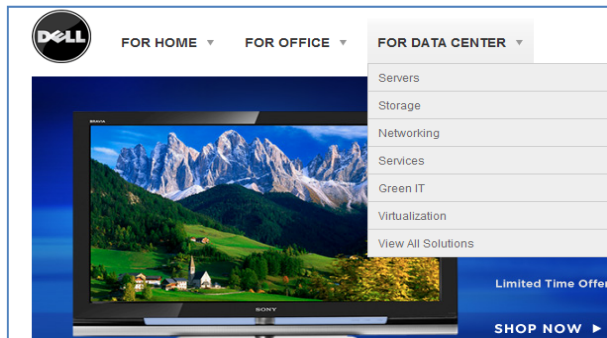

Figure 1. The Menu at Dell's Main Website

The students were then asked to do a lab exercise creating an HTML page with an ordered list of the menu items in VS 2008. They were instructed to create a new Web site in Visual Studio (File / New / Web Site, choose the default of ASP.NET Web Site), and then add an html file (Web Site / Add New Item / HTML Page) with the basic list and the sub-menu (the fly-out items) shown in Figure 2. This

ordered list is similar to the one used on Dell's website. The source html is shown in Appendix 1.
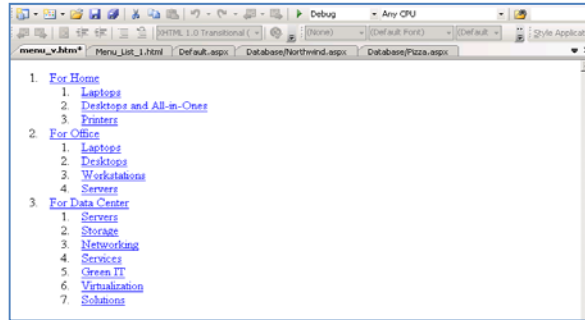

Figure 2. The Menu List

Hyperlinks were added into menu items for the navigation purpose. Since the purpose of this exercise is to demonstrate menu design, those items can be linked to any page such as to Dell's main home page. Moreover, the students were required to examine the *<ol>, *, and *<a>* HTML tags for the ordered list and the hyperlinks by using the Source View of VS 2008 (again, shown in Appendix 1).

At this point, students were excited to learn they could visually design a fly-out menu in CSS without a single line of code. They were then instructed to create a new CSS file (Web Site / Add New Item / Style Sheet) and attach it to the html page (by opening the html page in Design View, and dragging the Style Sheet, from the Solution Explorer, onto a blank space of the html page) so that they could add styles to the menu list. Adding these styles took place as follows:

Add Padding and Margin Style for the "ol" Selector: The dialog box shown in Figure 3 was invoked by choosing View / Manage Styles, and then clicking on the "New Style" hyperlink. As shown, "ol" was the Selector chosen, and the value "0" was assigned to both padding and margin in the "Box" category in CSS to remove the inherent styling of the ordered list. In addition, "none" was set as the value for the List-Style-Type in the List category in CSS. These two steps ensured that both main menu items and sub-menu items align with the left side of page without any indentation and can work the same with any current web browser (IE 7.0 or higher). Figure 4 shows the page after applying these styles.
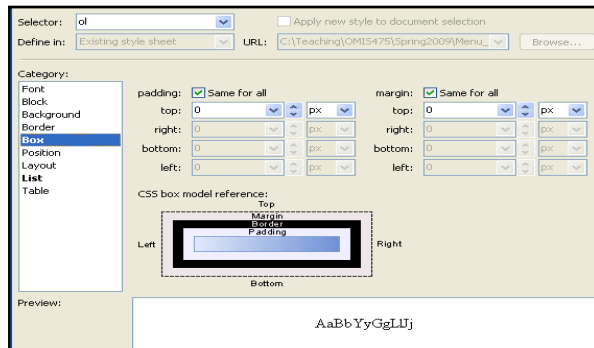

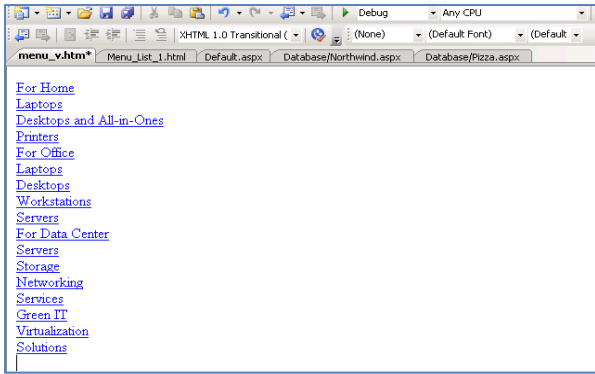**Figure 3. Reset the Padding & Margins in CSS**

Figure 4. Result for Resetting the Padding and Margin

Add Sub-Menu Styling for the "ol li ol" Selector: Next, the students learned to make the sub-menu flip. They created a new style for the "*ol li ol*" Selector in CSS to set the float attribute to "left" in the Layout category as shown in Figure 5. This changed the sub-menu display. To make the menu align horizontally, an additional style was defined for the "*ol li*" Selector (not shown, but similar to Figure 5). In this style, in the Layout category the display value was set to "inline" and the float value was set to "left", and in the Position category the width attribute was set to "160px" and position was set to "relative". Figure 6 shows the result after applying the new styles.
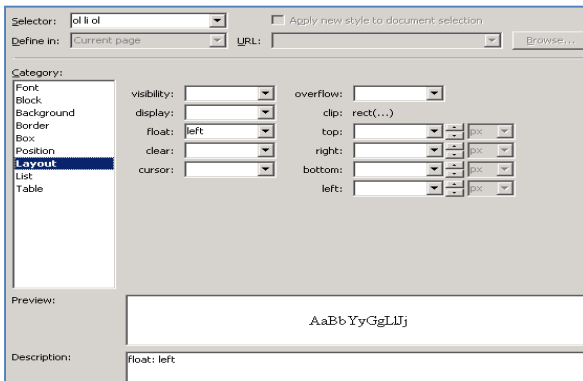

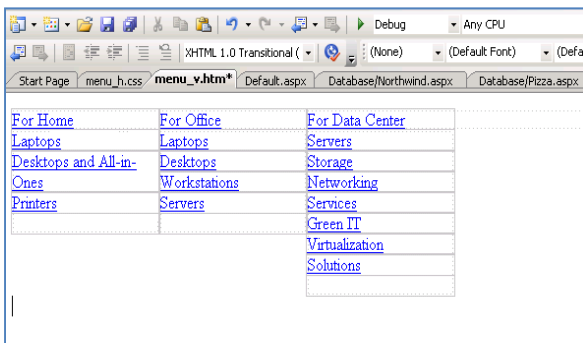**Figure 5. Styling the Sub-Menu Items**

Figure 6. Result for Styling the Sub-Menu

Add Menu-Hide and Hover Styles for the "ol li a" Selector: Next, the students learned to style the hyperlinks to make the menu display in a pleasing manner. They created a new style for the "*ol li a*" Selector in CSS to affect all links within the ordered lists. The students were given a chance to define their own font-family, font-size, font-color, border-color, border-style, border-width, and background-color (all in the "font" category) in this exercise. Moreover, "none" was set as the value under the text-decoration attribute in CSS (still in the "font" category) and "block" was chosen as the display value in the Layout category to remove the underline for each hyperlink to display a professional menu. Finally, "5 px" was set for the padding value in the Box category, and "15 px" was set for the height value in the Position category.

Add Menu Disappear and Reappear Styles for the "ol ol" Selector: The final step of the fly-out menu design involved making the sub-menu disappear and reappear when surfing the web. The students created a new style for the "*ol ol*" Selector in CSS to accomplish this task. The students learned to set the display value in the Layout category to "none", and the position value to "absolute", the top value to "25 px", and the left value to "0 px", all in the Position category (as shown in Figure 7). Since the "*ol ol*" style affects all sub-menus, this specification effectively hides the sub-menu from the screen as well as from mouse clicks.
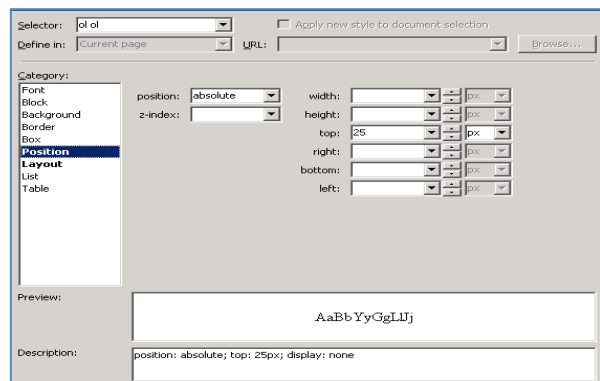

Figure 7. Hide the Sub-Menus

Add Menu Visible When Hovering Styles for the "ol li:hover ol, li" Selector: Figure 8 shows how to make the sub-menus visible again when hovering to the main menu. This style was realized in the "*ol li:hover ol, li*" Selector. The students learned that the *:hover* pseudo-class is needed on the first list item before the second ordered list. This style would only affect ordered lists that are found under list items that are currently being hovered over inside ordered lists. In addition, they were instructed to set the display attribute to "block" in the Layout category in CSS. Figure 9 shows the completed horizontally displayed fly-out menu.

4. CONCLUSIONS

This paper described using CSS in Visual Studio 2008 to design a fly-out horizontal menu for a website. Appendix 2 shows the complete CSS file for the menu format. Interested readers can also access the example at

http://ebusiness.cob.niu.edu/m10cxl1/menu_h.htm. Students were excited that the exercise helped them better understand how to apply and use CSS, especially in a visual environment using no code. CSS has become a recognized standard for professional Web design, but many students and faculty alike have not fully utilized it given the clunky, command-driven interfaces which are prevalent.
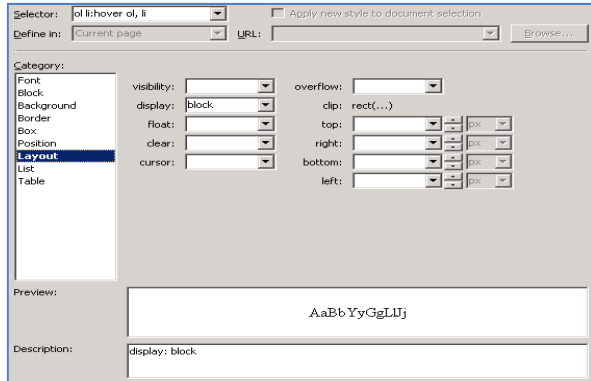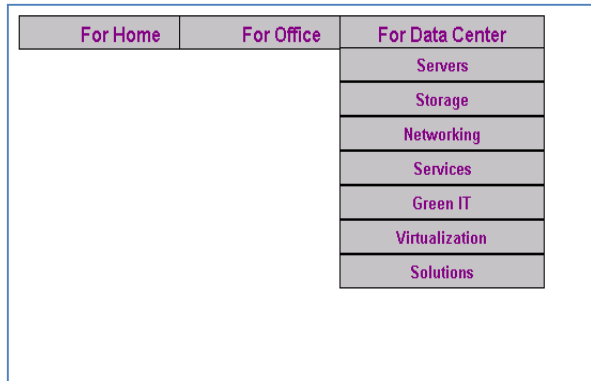

Figure 8. Make the Sub-Menu Visible in CSS

Figure 9. The Fly-Out Menu in Horizontal Display (currently the mouse is hovering over "For Data Center")

This exercise has presented to our students a very practical way to maintain and manage a website. The menu can be easily amended and expanded since it is based on an ordered list as shown in Figure 2. Students may simply change the list items themselves to revise the menu as a whole. Since the menu originally created might be changed or expanded as a site grows, this exercise helped students understand how using CSS can make a site design and/or redesign less invasive and time consuming. Completing this simple, canned exercise has shown our students that CSS is not only powerful, but also usable. They were able to see how CSS has the capacity to create high-quality interfaces with minimal effort. Realizing this high results/effort ratio of CSS within Visual Studio has motivated our students to learn more and push farther with CSS.

For example, the students were further challenged to modify the style in CSS to display the fly-out menu vertically as shown in Figure 10. They learned that to change the menu from a horizontal to vertical display they only needed to change the value of "inline" to "block" for the display attribute of the Layout category in the "*ol li*" Selector. No further changes are needed to achieve this vertical effect.
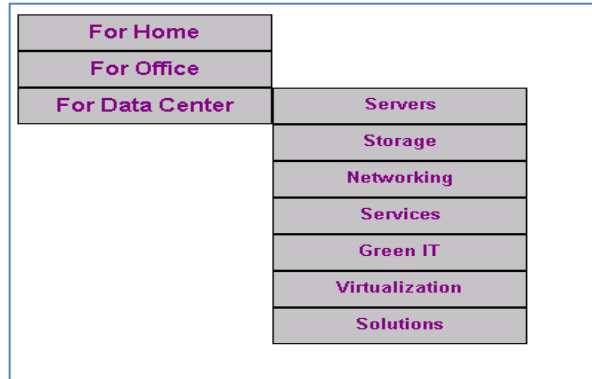

Figure 10. The Fly-Out Menu in Vertical Display

Using this exercise as a springboard, we believe it is possible to help students "learn how to learn" CSS, thus encouraging them to produce high-quality websites in less time. We feel other faculty can have similar results, and the following are some suggestions and additional ideas for teaching similar projects in web design classes:

- "Forcing" the students to complete this exercise will show the students how powerful CSS can be. Once completed, we have found that students view exercise as "easy", and are willing to experiment with different fonts, formats, etc., and this has increased their overall confidence with CSS. To evaluate student understandings for this exercise, they should be required to complete a similar assignment or exercise of a menu design with different fonts, sizes, colors, backgrounds, and text alignments.

- Instead of only using Visual Studio (VS) 2008 in this exercise, it would be very helpful to ask students to extend this exercise using Microsoft Expression Web (EW) to enhance their learning of using a CSS visual tool. For example, the instructor could use the CSS tools within VS 2008 to teach students to design a horizontal menu, and then require them to use the CSS tools within EW to design a vertical menu.

- It would also be helpful to change the ordered list to an unordered list for the menu design. Figure 2 shows an example of using the <ol> tag in this teaching tip to format the menu design. The instructor could instead require the students to use

the <ul> tag to format a menu display using bullet items. Here, the instructor could use Figure 2 as the HTML template to guide students to complete the CSS design. The instructor could then revise this HTML template to unordered lists with <ul> tags. Students would then have to complete their own CSS designs based on those unordered lists. This method would enforce the concepts of CSS design without having the students write a single line of code.

- Students should be challenged to further explore using CSS to design multi level sub-menus. The example used in this paper employs only one level of sub-menu. However, it can be easily extended to include multiple levels of sub-menus. In the future, we would like to require students to include a multi level sub-menu design for their final project. We believe it can lead to a better understanding of using CSS.

We are confident that using these suggestions, along with the CSS example described in detail in this teaching tip, will prepare students to design usable pages for their future employers, with minimal effort.

5. REFERENCES

Accessibility Features of CSS. (1999), Retrieved from http://www.w3.org/TR/CSS-access

Baxley, B. (2003), Making the Web Work: Designing Effective Web Applications. New Riders Publishing, Indianapolis, IN.

Downing, C. and Liu, C. (2010), "Does Web Site Usability Correlate with Web Site Usage?" Journal of International Technology and Information Management (JITIM), forthcoming.

Duckett, J. (2008). Web Programming with HTML, XHTML, and CSS, Wrox Wiley Publishing. Indianapolis, IN.

Internet Usage Statistics. (2010) Retrieved from http://internetworldstats.com/stats.htm.

Krug, S. (2006). Don't Make Me Think, Second Edition: A Common Sense Approach of Web Usability, New Riders Publishing, Indianapolis, IN.

Nielsen, J. (200). Design Web Usability. New Riders Publishing , Indianapolis, IN.

Spaanijaars, T(2008). Beginning ASP.Net 3.5 in C# and VB. Wrox Wiley Publishing, Indianapolis, IN.

AUTHOR BIOGRAPHIES

Chang Liu, DBA, is a Professor of MIS and Department Chair at Northern Illinois University. He received his doctor of business administration from Mississippi State University in 1997. He teaches database and electronic commerce courses and received MIS Graduate Teaching Awards in 2002, 2003, and 2006. His research works published *at Information & Management* (I&M), *International Journal of Electronic Commerce and Business Media*, *Journal of Global Information Management*, *Journal of Internet Research*, *Journal of Computer Information Systems*, *Mid-American Journal of Business*, *International Journal of Mobile Communications*, *Journal of International Technology and Information Management*, and *Journal of Informatics Education Research*.

Charles E. Downing, Ph.D, is the Presidential Teaching Professor in the Operations Management and Information Systems Department of the College of Business at Northern Illinois University. He received his Ph.D. from Northwestern University. He researches, teaches and consults in topics such as Systems Development, Systems Analysis and Design, Information Technology as a business process integrator (Supply Chain and portals), and Business Intelligence. His articles have appeared in *Decision Sciences*, *Communications of the ACM*, *Decision Support Systems*, *Information & Management*, *Journal of Global Information Management*, *The Journal Information Technology Management*, *Information Systems Management*, *Journal of End User Computing*, *Journal of Educational Technology Systems*, *European Journal of Operational Research*, and *Journal of the Operation Research Society*.

Appendix 1: The Source html for the Main Page

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>



  <link href="StyleSheet.css" rel="stylesheet" type="text/css" />


  </head>
<body>

  <ol>
    <li><span lang="en-us"><a href="http://www.dell.com">For Home</a></span><ol>
      <li><span lang="en-us"><a href="http://www.dell.com">Laptops</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Desktops and All-in-Ones</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Printers</a></span></li>
      </ol>
    </li>
    <li><span lang="en-us"><a href="http://www.dell.com">For Office</a></span><ol>
      <li><span lang="en-us"><a href="http://www.dell.com">Laptops</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Desktops</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Workstations</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Servers</a></span></li>
      </ol>
    </li>
    <li><span lang="en-us"><a href="http://www.dell.com">For Data Center</a></span><ol>
      <li><span lang="en-us"><a href="http://www.dell.com">Servers</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Storage</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Networking</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Services</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Green IT</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Virtualization</a></span></li>
      <li><span lang="en-us"><a href="http://www.dell.com">Solutions</a></span></li>
      </ol>
    </li>
  </ol>

</body>
</html>
```

Appendix 2: The CSS File for a Horizontal Menu Display

```css
ol
{
   padding: 0px;
   margin: 0px;
   list-style-type: none;
}
ol li ol
{
   float: left;
}
ol li a
{
   font-family: Arial, Helvetica, sans-serif;
   font-weight: bold;
   color: #800080;
   text-decoration: none;
   text-align: center;
   background-color: #C0C0C0;
   border: 1px solid #000000;
   padding: 5px;
   width: 160px;
   height: 15px;
   display: block;
}
ol ol
{
   display: none;
   position: absolute;
   clip: rect(0px, auto, auto, 0px);
   top: 25px;
   left: 0px;
   font-size:smaller;
}
ol li:hover ol,li
{
   display: block;
}
ol li
{
   position: relative;
   float: left;
   display: inline;
}
```

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.