

The following assignments all feed into one final deliverable - a contribution to an ongoing tutorial to help first and second year SE (and CS) students learn how to increase the usability of their program's GUI and interaction. With a partner, you will select a GUI element/interaction style and a technology. For that pairing, you will develop a tutorial that would be suitable for a 2nd year student to use in a course such as SE 361. While you will need to do research (cite those sources), the work must be your own as the tutorial will be available on the web for students to use. You need to select your GUI element/interaction style and technology selection on the Assignment 2 Discussion group, but the selection must be approved by the instructor. Suggestions include:

*Technology:* SwingX (Java), C++, Flash, Silverlight, C#/.net **Note:** Javascript, html can be a part of a solution but not the solution

*GUI element/interaction style:* e.g. drag and drop, tree-style browsing and item selection, JXTable (SwingX) with highlighting and sorting of info Be creative and think about useful (nontrivial) ways people interact with software.

### **Assignment 3: Tutorial Topic - Proof of Concept 'App' and List of Resources (150 pts)**

The goal is that the interaction style and/or the technology selected requires you and your partner to learn something new as a way of enhancing your skills. As part of the tutorial, you need to create a small application that demonstrates the proof-of-concept of the interaction/gui element in the technology that you chose. The application should be small but have some use to the audience. Feel free to be creative, though it should be your own work.

As you think about the tutorial and go through the process of creating the proof-of-concept app, you will also need to gather resources that you feel will be helpful to you and to the prospective audience in terms of learning about the technology for your selected gui element/interaction. Keep track of these resources as you will include them in your assignment here and in the tutorial as a whole. You need to include the title of any web resources used along with their urls. Each resource item should have a brief description in terms of why you selected it (its value).

*What is turned in: For each pair, submit the code and packaged version of your proof of concept app. In addition, submit a list of at least 5 resources that you intend to use in your learning of the concept/technology and that you intend to list for the reader to 'learn more about it'. Be prepared to give a demo in class.*

Grade is derived from:

- finished proof-of-concept app that is clear and demonstrates the selected ui elements/interaction style: 85 pts
- code for app is clear and commented: 30 pts
- readme is included with any instructions needed to run the app: 15 pts
- List of resources, with titles and bibliographic info or urls: 20 pts

NOTE: Major penalties are if the app is not finished, doesn't do anything (other than showing what the UI looks like), there is no source turned in, the app doesn't run, the app isn't provided in a compiled version (if applicable), etc.

#### **Assignment 4: UI Tutorial Contribution (125 pts)**

The outline for your tutorial is in mycourses. You should use the provided headings where possible, some addition to them may be needed. Include graphics where appropriate - to provide value in the tutorial. The graphics must be inline or linked from the write-up - if you are linking to the graphic, include a 'thumbnail' or smaller version of the graphic that can be displayed inline. The write-up must be in HTML though not the HTML created in Word (it is too messy).

You will be linking to your proof-of-concept 'app' from your webpage. If applicable, your proof-of-concept app can be inline, otherwise a link is suffice. Remember these tutorials will be online so be sure to proofread.

*What is turned in: For each pair, submit the tutorial as a zipped directory that contains the tutorial file(s) in HTML and any image and code files. Make sure any links within your own files are relative links.*

Grade is derived from:

- Finished tutorial (write-up/app paring) is structured and written at the appropriate level for the prospective audience (novice programmers such as high school or intro students): 70 pts
- Tutorial write-up is clear, well-written (e.g. good technical writing practices, grammar, spelling), is in HTML, and includes inline code snippets to help explain the topic of your tutorial: 40 pts
- Finished app 'screens' and other graphics are inline, including any needed thumbnails: 15 pts.
- Template is used, with any appropriate customizations, including the list of resources, with titles and bibliographic info or urls; cite any quoted material: 15 pt penalty if this the template is not used

NOTE: Major penalties are if the write-up is not finished or is poorly written, the write-up does not correspond to the app and/or topic, the write-up is not in HTML and the graphics are not inline, the tutorial consists of just the code (e.g. It is self-explanatory), there are no resources, etc.