

Instructor: Kenny Wong

Final Exam

CMPUT 301 : User Interfaces and Software Design **A3**

December 11, 2000

Identification

Last Name: _____

First Name: _____

Student ID: _____

Instructions

Duration: 180 min

Weight: 35%

Read all the instructions.

The exam consists of 9 pages, including this cover page, and contains 8 questions worth a total of 70 points. The number of points assigned to each question is shown in square brackets. Please check that you have received a complete exam.

All questions are to be answered in this exam. This is an open-book, open-notes, but closed-neighbor :) exam.

Use the backs of pages if necessary for additional working or answering space.

Start only when instructed. The time remaining will be indicated and updated at about 10 min intervals.

Good luck!

Scoring

Question	Mark	Value	Question	Mark	Value
1		/10	5		/10
2		/10	6		/10
3		/4	7		/8
4		/8	8		/10
			Total		/70

E

05176

WONG, K.
 DEC 00 FINAL
 PAGES: 9
 CMPUT 301 (A3)

NA

Questions

1 Experimental design

Design a controlled experimental study to evaluate the effectiveness of two Java integrated development tools: CodeWarrior and VisualAge/Java. The tasks (and features) of interest are browsing and editing source code.

- (a) [2] You have 30 users of varying programming and Java experience who have provided the appropriate consent to participate in the study. They have never used the tools.

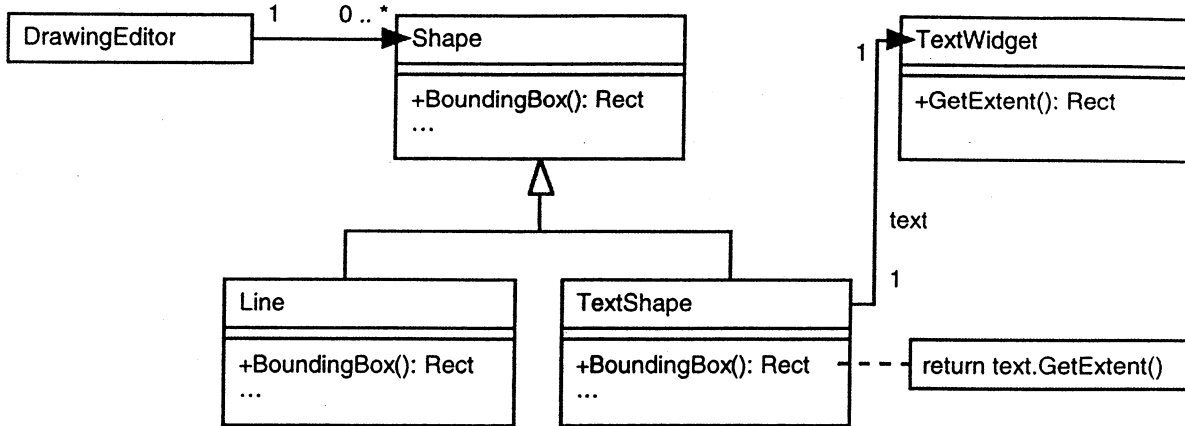
Describe how you would assign the users to the tools. Are you using a between-groups or within-groups experimental design? Why?

- (b) [8] Also, each experimental session with a user can take a maximum of 2 hours.

Describe the major activities and combination of evaluation techniques you would employ. Indicate, where relevant, the time duration allotted for each part of the session. List any assumptions you need (e.g., representative bodies of source code).

3 [4] UML notation

Study the following UML class diagram carefully. This is an instance of a common design pattern that is being used in a drawing editor.



Explain what the intent or purpose of this pattern apparently is. You do not need to name the pattern.

4 [8] Models of the user

Describe how you would try to predict the relative speed of access of the following two ways of accessing an application's full set of menu items:

- pull-down menus in a menu bar anchored at the very top of the screen
- pop-up menu at the mouse pointer with hierarchical submenus.

State your assumptions.

5 [10] Task analysis

Use the task analysis for knowledge description technique to create a task descriptive hierarchy over the following objects:

CD recordable drive, DVD drive, flatbed scanner, game pad, hard drive, inkjet printer, joystick, keyboard, laser printer, microphone, modem, monitor, mouse, speakers, tape drive,

Make sure your taxonomy satisfies the uniqueness rule. Label categories with AND, XOR, or OR as appropriate.

6 Dialog notations

Consider the interaction of dragging and dropping a document icon over a target icon (e.g., an application). The document icon moves with the pointer during the drag. Also, the target icon highlights when the pointer moves directly over it and if it can accept the document.

(a) [4] Specify an EBNF grammar for this interaction. State your assumptions.

(b) [6] Draw a UML state diagram for this interaction.

In the diagram, also account for the situation where the user decides not drop the document on the target. Label the elements of the diagram clearly. State your assumptions.

7 [8] Usability guidelines

In your own words, write a short essay describing four usability guidelines. Choose what you think are four important guidelines to know when designing graphical user interfaces. Assume your audience is a 3rd year computing science student who has not taken CMPUT 301.

8 [10] User interface design

The Unix `diff` command displays line-by-line differences between pairs of text files.

The command has the form:

```
diff options file1 file2
```

where the options can be a sequence of zero or more of the following four flags:

- b ignore trailing spaces and tabs
- i ignore case
- t expands tab characters in the output
- w ignores all spaces and tabs

(followed by zero or one other flags which will not be of concern in this design).

Example files and output:

```
file1.txt:      file2.txt:      diff file1.txt file2.txt:
this is        and            0a1
file1          this is       > and
               file2            2c3
                                   < file1
                                   ---
                                   > file2
```

That is, the second file has an added line at the top and a changed line. In general, differences can be additions, changes, or deletions.

Sketch the layout of a graphical user interface with the same essential capabilities and options as `diff` with an improved, more readable output presentation. Hint: use color.

