

Name \_\_\_\_\_  
Student ID Number \_\_\_\_\_

CSE 5324  
Software Engineering I  
Exam 2, Fall 1999  
© DL, UTA, 1999

GRADE: \_\_\_\_\_/100

**Please** read this:

This is a closed book, closed notes exam. You may use a single sheet of notes **which must be handed in with the exam.**

Use a dark ink (or pencil) and print answers on the test paper following the question or on separate paper. Please **do not write on the back of sheets** of paper **or** on the **back of the exam.** Please **keep answers to questions together** and try to avoid continuing answers off the page.

Turn in all pages of the test. **Write** your answers **legibly.** Unreadable answers will be counted wrong. Make sure you have all pages of the test.

Read each question carefully (twice) and be sure your answer addresses the question. Overly general (non-specific) answers will be counted wrong. If any part of an answer is incorrect (even if other parts are correct) points will be deducted.

Point values are given for each question. The exam has a total of 100 points.

Please turn in the exam promptly when called for. **Late exams will have points deducted.**

Because of limited time the instructor will **not** be able to **answer questions** for the last **five minutes** of the exam time.

Points deducted for:

- [            ] Writing on back of pages.
- [            ] Not turning in "reminder sheet" ("cheat" sheet)
- [            ] Disjoint (non-contiguous) answers
- [            ] Late exam

1. [10 pts]

Short answers:

a.) Briefly (in a few words) what is the relationship between the following (OO) terms (in 5 to 10 words describe, and give relationship):

- 1.) A method
- 2.) An object
- 3.) Polymorphism
- 4.) Encapsulation
- 5.) UML

2. [40 pts]

The following merges two lists, both of which are contained in one array: "lists". The first list starts at location (index) "first1" and goes through "last1", the second list starts at "first2" and goes through "last2".

For the following code fragment:

```
1. void merge (int lists[], int first1, int last1, int first2, int last2)
2. {
3.     int temp[MAX_ARRAY];
4.     int index, index1, index2;
5.     int num, num_in_1, num_in_2;
6.
7.     index = 0;
8.     index1 = first1;
9.     index2 = first2;
10.    num = last1 - first1 + last2 - first2 + 2;
11.    while ( (index1 <= last1) && (index2 <= last2) )
12.        {
13.            if (lists[index1] < lists[index2])
14.                temp[index++] = lists[index1++];
15.            else
16.                temp[index++] = lists[index2++];
17.        }
18.    if ( index1 > last1 )
19.        move( lists, index2, last2, temp, index );
20.    else
21.        move( lists, index1, last1, temp, index );
22.    move( temp, 0, num-1, lists, first1 );
23.
24. }
```

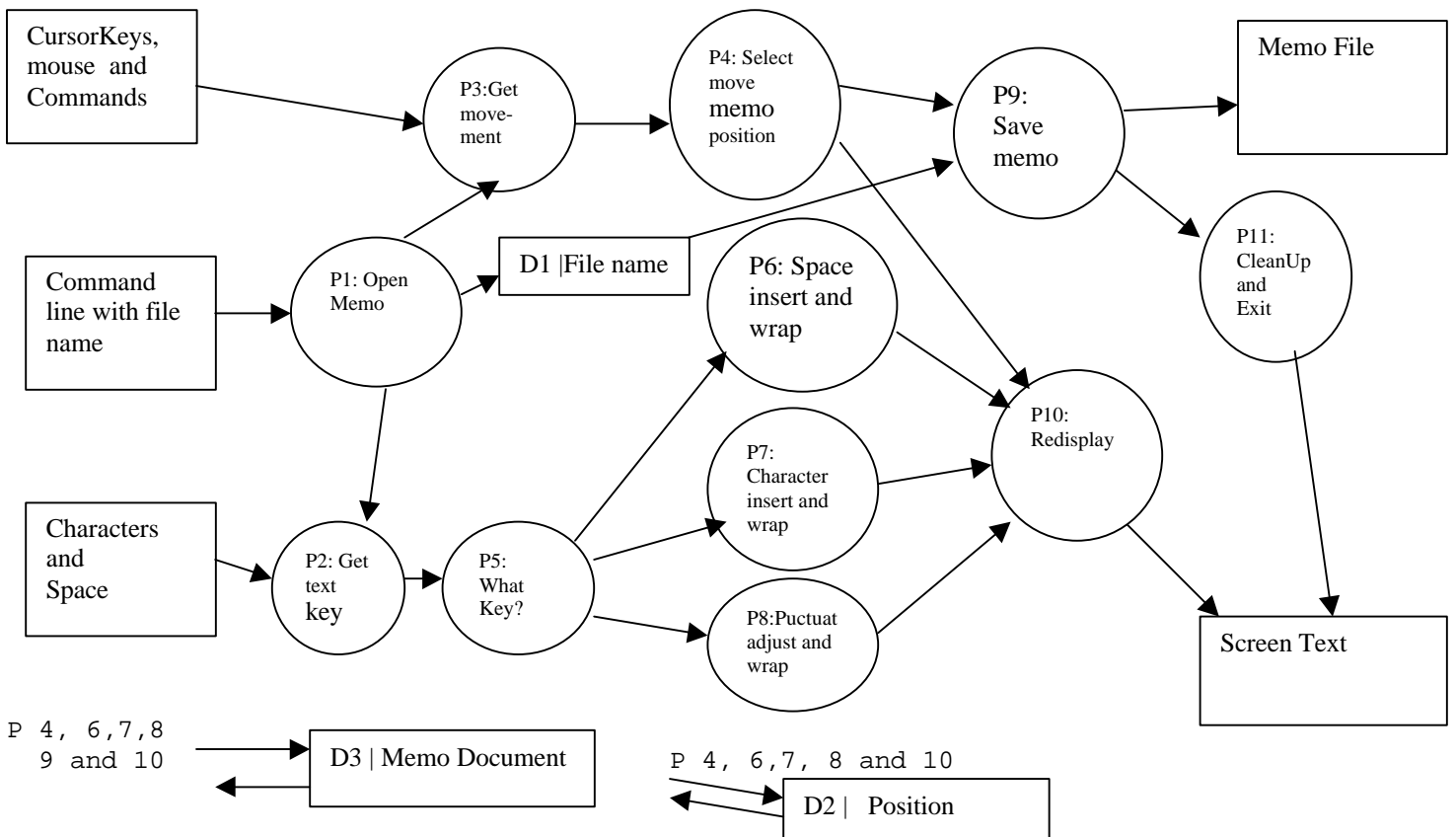
Note: There are no intentional syntax errors, but there may be errors in this implementation. **When asked for a testing methodology, be specific (for example: this is a blue-box, Independent-Loop test) If more than 1 possible, list at least 2.**

- a.) 1.) Give a (simple) black box test example for this program.  
Can this be written without looking at the code? How?
- 2.) What does a black box test actually test (what type of errors are found)?  
How would one know if the test was successful?
- b.) Draw the corresponding flow graph. (Not here!)
- c.) How many linearly independent paths are there through the flow graphs (or program)?  
(Show how you calculated this) and give 3 independent paths (if less, list all).
- d.) Show test cases that will force the execution of the following lines:  
(Show actual values: like "[0 1 2 3], 0, 1, 2, 3" to test with)  
Line 16  
Line 19
- e.) What type of testing is done by "def-use" or "all-use" testing? Give an example using "index1".
- e.) How would a "code inspection" (Fagan) be done for this routine? Explain.  
Would any other type of "inspection" be valuable, in this case? Explain.
- f.) When (specifically, at what stage of development) and how (if you wish you may give an example) would one use "stubs" or "drivers" with this merge routine?
- g.) One big issue is "when to stop testing". Please give one metric (measurable) criteria to determine when testing is "complete".

3. [40 pts]

Your company, MemoWriter, wants to create a one page memo writing program. The one page memo consists of text (which are ASCII printable characters and punctuation (".", ",", and "?") ). These characters form words, grouped into sentences, which form paragraphs. Text lines are word-wrapped so that a line does not exceed 72 characters, for all the lines within a paragraph. Punctuation is always "attached" to the preceding word (no space allowed) even if the user did not enter it that way. The "cursor" (insertion point) may be moved by the cursor keys or mouse, and the memo may be saved by typing "Control-S".

The following DFD represents MemoWriter.



P 4, 6, 7, 8  
9 and 10

P 4, 6, 7, 8 and 10

( Note D2 and D3 have many processes that use them, shown above for simplicity)

- a.) Show a design mapping: show a structure chart diagram (design) for this DFD. The first (highest) module should be a "driver module called "M". Portions of this DFD may be transform or transaction. Show which modules correspond to which processes. (Use the same numbers, M1 represents P1) All processes should appear as at least one module. Very briefly explain how and why you created your design.

- b.) The M-spec for M4 is:

```

    Command or movement (M4):
    If ( key = Control-S ) then go to save
    else
        If (key = left-cursor or key = right-cursor)
            then move-inline-cursor( key )
        else
            move-betweenlines-cursor( key )
    
```

- 1.) Describe any types of coupling and cohesion you can identify (for M4). Be specific - what types and why (give example)

- c.) Some (OO SE's) would say this is clearly better if done using OOA and OOD. Assuming the specification above, describe (use UML, text book, class, or any similar notation) the object "Memo Document" (D3). Are there any more objects identifiable? If so, which, and briefly describe them.
- d.) In the your structured design in part a.) how do the arrows (flows), processes, and data get used in your design (what is the mapping)?
- e.) If you were using a "top-down" integration how would you integrate and test MemoWriter? (Give details)

Bonus:

[5]

What are Brooks recommendations regarding: design, integration and testing?