

The University of Texas at Dallas
Computer Science Program

Test 1

October 16, 1997

Conditions: Closed book Duration: 70 minutes

Name:

_____ {Please underline last name}

Student Number:

1. _____ /20

2. _____ /20

3. _____ /20

4. _____ /20

5. _____ /20

Total _____ /100

1. [20 marks]

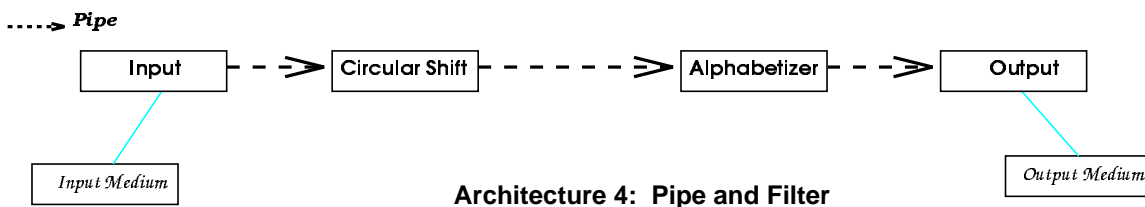
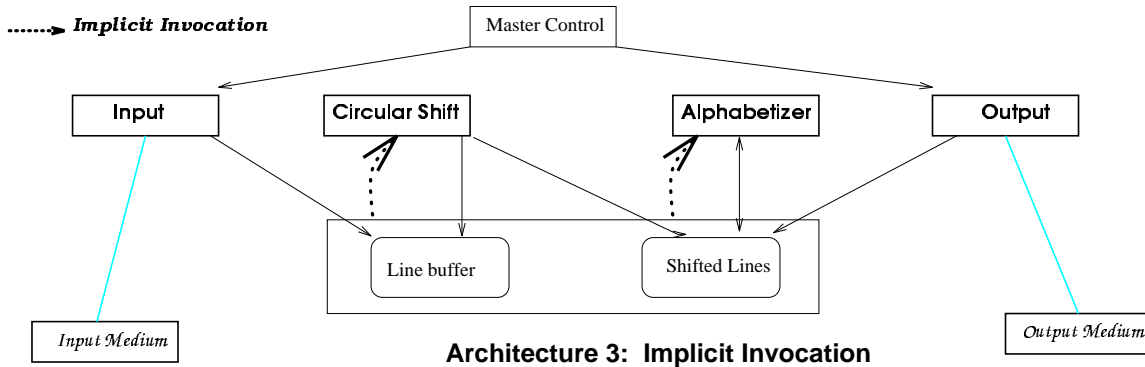
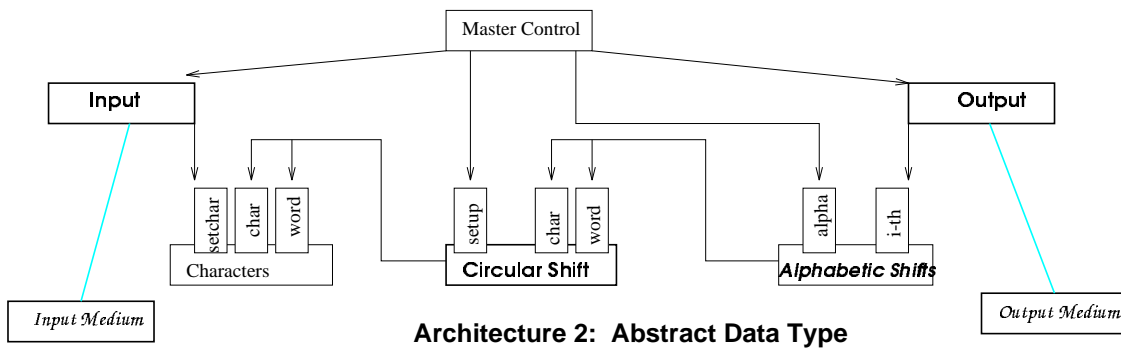
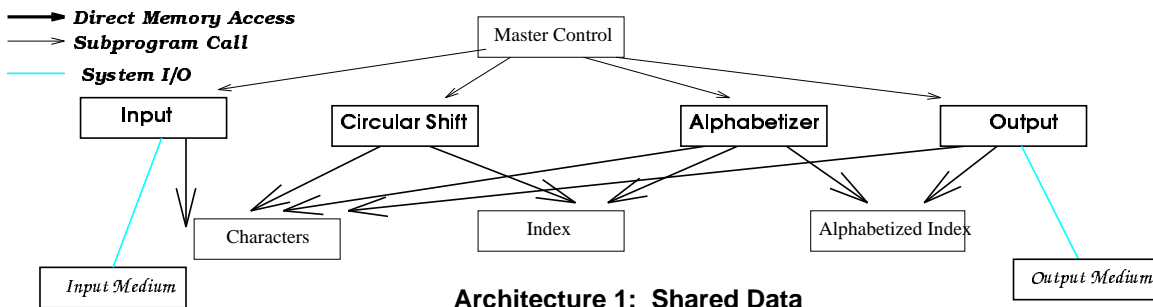
page 2/8

For each of the following ten statements, indicate whether it is true (mark T) or false (mark F).
(No penalty for a wrong answer)

 T This is the Test 1 for CS6362.

- ___ 1. The use of a Structure Chart (Program Control Hierarchy) cannot result in a layered system for, say a traffic control system.
- ___ 2. In the context of telephony system architecture, the correctness of a client Object is independent of the correctness of its server Object.
- ___ 3. The architecture of an Object Management System should be quite similar to that of a relational Database System, as an Object corresponds to a tuple.
- ___ 4. The quality of a software architectural design largely depends on the quality of the process whereby the design is produced.
- ___ 5. As is the case with Unix operating system, the essence of a pipe-and-filter architecture can be specified using C.
- ___ 6. In building a critical system (e.g., a patient monitoring system), a software architect should select only those software architectural designs that completely guarantee the satisfaction of all essential non-functional requirements such as reliability.
- ___ 7. In the style of Shared Data, modules can communicate with each other partly via procedure calls.
- ___ 8. In an ADT architecture, modules typically communicate with each other via (remote) procedure calls and asynchronous message passing.
- ___ 9. An essential part of any software architectural design should be design rationale, since design rationale explains why the particular architecture is chosen from the (possibly infinitely) large design space..
- ___ 10. A semi-formal approach to designing a software architecture is often considered bad concerning defects but good concerning understandability.

Consider the following four architectures for the KWIC problem.



2.1 Consider *Architecture 4*. Describe briefly what and where modification is needed to efficiently “omit” indices starting with a noise word (e.g., the, a, an, to, and, or, etc.).

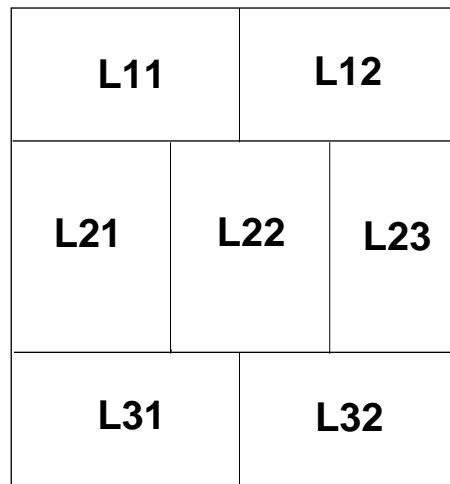
2.2 Consider *Architecture 1* and *Architecture 4*. Compare them with respect to reusability and space performance.

2.3 Consider *Architecture 2*. Suppose the **Master Control** module is to be deleted from the architectural design. Describe briefly What should be done to achieve the same functionality as before?

2.4 Consider *Architecture 1* and *Architecture 3*. Describe briefly the major disadvantage(s) of **Architecture 3**, when compared to *Architecture 1*.

3. [20 marks]

Consider the following figure given as the layered software architecture for a software system, called *System Integrator*.



4.1. Although the details of what each module (e.g, L_{ij} , $i, j = 1, 2, 3$) does are unknown, a software architect can point out certain constraints associated with this style of software architecture. Describe in a point form what such constraints are.

4.2. Aside from the issue of formal semantics, this figure is not quite adequate even as a diagrammatic software architecture. Describe in a point form major deficiency (ies) of the figure as a software architecture.

3. [continued]

page 6/8

Suppose the software architects of System Integrator have proposed another software architecture for the system, this time based on pipe-and-filter. Also suppose that they have provided an analysis of tradeoffs between the two software architectures, as given in the table below:

	Layer	Pipe & Filter
Modifiability	-	+
Performance	+	-
Reusability	+	+
Flexibility	+	+
Testability	-	+
Traceability	-	+
Usability	+	-

4.3 Discuss who would be involved in the analysis and why.

4.4 List and discuss major issues with carrying out a tradeoff analysis in the manner above.

4. [20 marks]

page 7/8

Consider the following declaration of stack:

```
Stack (E, C): trait /* E(e.g., integer, string) is an element of C, a stack */
  introduces
    new: -> C
    push: C, E -> C
    top: C -> E      exempting top(new)
    pop: C -> C      exempting pop(new)
    isEmpty: C -> Bool
  asserts
    C generated by new, push
    forall stk: C, e: E
      top (push(stk, e)) == e
      pop (push(stk, e)) == stk
      isEmpty(new)
      ~ isEmpty(push(stk, e))
  implies
    LinearContainer (push for insert, top for first, pop for rest)
```

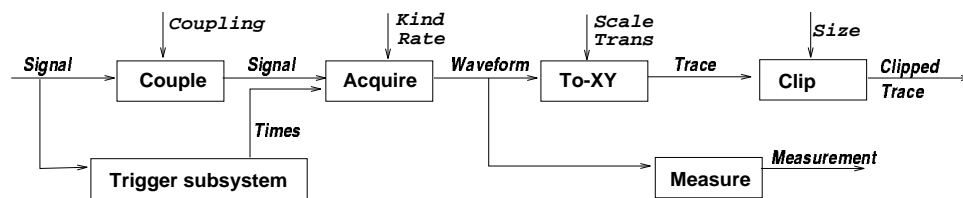
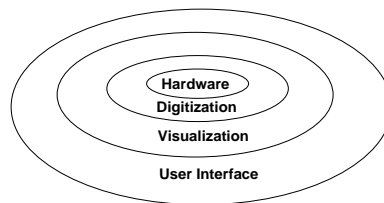
Now, suppose you are using a theorem prover, called *TP*. Also suppose that “Alice” and “Wonderland” are elements of *E*.

1. What should *TP* return as the value of `top (push (new, “Alice”))`?
2. What should *TP* return as the value of `pop (push (new, “Alice”))`?
3. What should *TP* return as the value of `isEmpty (pop (push (new, “Alice”)))`?
4. What should *TP* return as the value of `pop (push (push (new, “Alice”), pop (push (new, “Wonderland”))))`?
5. What should *TP* return as the value of `insert (rest (insert (new, “Alice”)), “Alice”)`, assuming that `LinearContainer` has access to `Stack`?

5. [20 marks]

page 8/8

Consider the following two architectures for an oscilloscope, as discussed in class:



1. Suppose Architecture 1 is chosen instead of Architecture 2. For each component of Architecture 2, indicate what layer it would belong to.

2. What kind of mechanism would be needed to display **Measurement** on the screen?