The University of Texas at Dallas
Computer Science Program

Midterm Test March 6, 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]

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 mid-term test for CS6362.

_*F*_ 1. All compilers run in a batch mode as this mode is good for conceptual simplicity and adaptability.

_*T*_ 2. The correctness of a filter is independent of the correctness of its predecessor filters.

_*T*_ 3. A software architecture serves as an (abstract) skeleton which can be used to expose the ability of a system to meet its gross system requirements.

_*F*_ 4. Programming-in-the-small focuses on building evolvable software systems, while programming-in-the-large focuses on building efficient data structures and algorithms.

_*F*_ 5. Classical module interconnection languages (MILs) are powerful enough to describe a variety of architectural styles, such as pipe-and-filter and inplicit invocation.

_*F*_ 6. The single primary role of non-functional requirements during software architectural design should be in selecting among software architectural alternatives after they have been produced by software architects.

_*F*_ 7. In the style of implicit invocation, modules communicate indirectly with each other by directly accessing shared data.

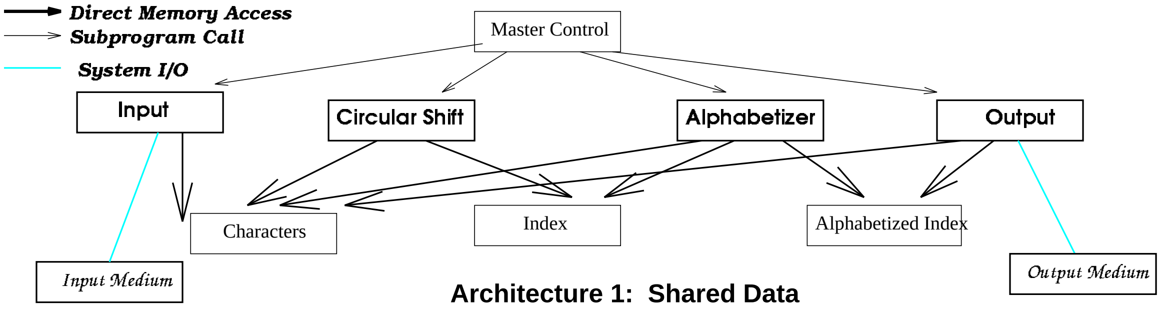
_*F*_ 8. In a batch sequential architecture, data flows through a sequence of discrete processing steps where update modules can run concurrently with each other.

_*T*_ 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.

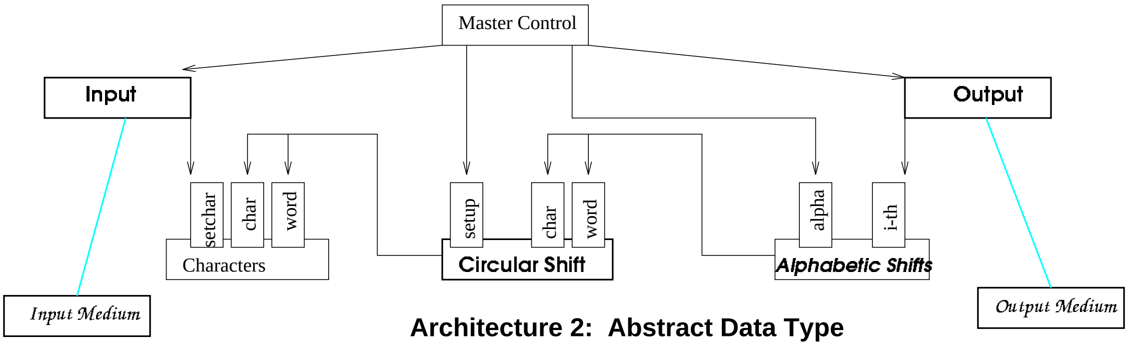
_*T*_ 10. A semi-formal approach to designing a software architecture is often considered bad concerning defects but good concerning understandability.

2. [20 marks]

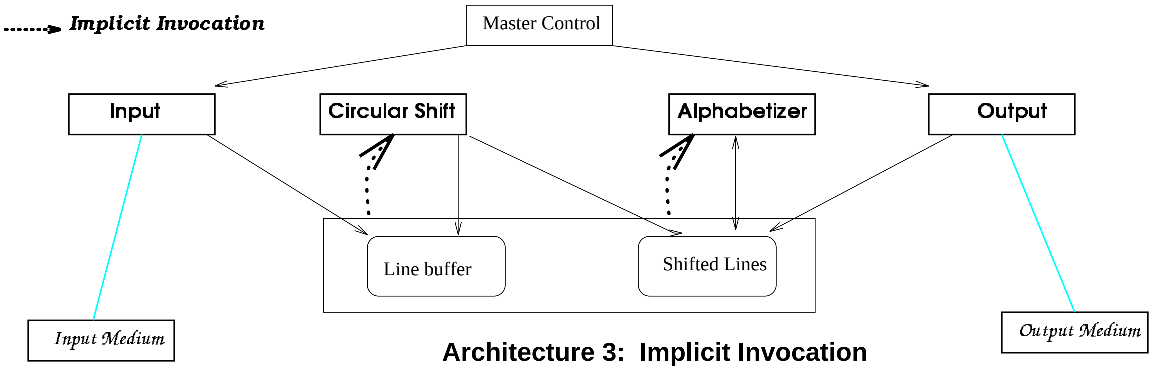
Consider the following four architectures for the KWIC problem.



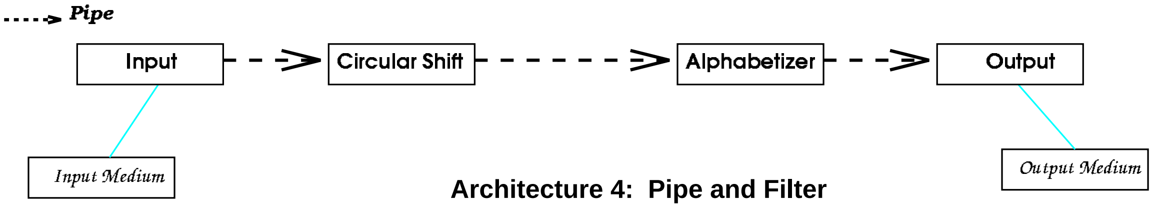
Architecture 1: Shared Data



Architecture 2: Abstract Data Type



Architecture 3: Implicit Invocation



Architecture 4: Pipe and Filter

Consider *Architecture 1*. 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.).

After the circular shift; a new "Index-without-noise" will be created and accessed by the Alphabetizer.

Consider *Architecture 1* and *Architecture 2*. Compare them with respect to reusability and space performance.

+ reuse is better supported in Architecture 2 than in Architecture 1, as modules make fewer assumptions about the others with which they interact

- space performance can be poorer in Architecture 2 than in Architecture 1, if not incremental

Consider *Architecture 3*. Suppose the Output module is to be implicitly invoked, instead of being explicitly invoked by the Master Control module. What kind of data should be generated, and by which module and when?

The Alphabetizer should generate, when it is done with sorting all the circularlly shifted lines, "end-of-alphabetization" data.

Consider *Architecture 1* and *Architecture 4*. Describe briefly the major disadvantage(s) of Architecture 4 when run in a batch mode, when compared to *Architecture 1*.

- inefficient use of space, as each filter must copy all of the data to its output ports (e.g., all circular shifts in full).

3. [20 marks]

Consider the following module declaration:

```
module M
       provides: a, b, c, d, e;
       requires: v, w, x, y, z;
       consist-of: module M1, module M2, module M3

       module M1
              provides: a;
              requires: v;
              string a, real v
       end M1

       module M2
              provides: b, c, d;
              requires: w, x, y;
              has-access-to: module M1
              consist-of: module M21, module M22, module M23

              module M21
                     provides: b;
                     requires: w;
                     boolean b, integer w
              end M21

              module M22
                     provides: c;
                     requires: x;
                     has-access-to: module M21
                     integer c, real x
              end M22

              module M23
                     provides: d;
                     requires: y;
                     has-access-to: module M22
                     boolean d, string y
              end M23
       end M2

       module M3
              provides: e;
              requires: z;
              has-access-to: module M2
              integer e, z
      end M3
end M
```

3. [continued]

1. List the set of (both internal and external) variables that module M1 has access to.

   a, v

2. List the set of (both internal and external) variables that module M21 has access to.

   b, w

3. List the set of (both internal and external) variables that module M22 has access to.

   c, x, b, w

4. List the set of (both internal and external) variables that module M23 has access to.

   d, y, c, x

5. List the set of (both internal and external) variables that module M3 has access to.

   e, z, b, c, d, w, x, y

4. [20 marks]

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)
```

1. What is the value of pop (top (stk)) for any stk: C?

   undefined

2. What is the value of top (push (new, e)) for any e: E?

   e

3. What is the value of pop (push (new, e)) for any e: E?

   the stack generated by ”new

4. What is the value of isEmpty (pop (push (new, e))) for any e: E?

   true

5. What is the value of pop (push (push (new, e), e’))) for any e, e’: E?

   the stack generated by ”new” containing e

6. What is the value of isEmpty (pop (push (push (new, e), e’))) for any e, e’: E?

   true

7. What is the value of top (pop (push (push (new, e), e’))) for any e, e’: E?

   e

8. What is the value of top (pop (pop (push (push (new, e), e’)))) for any e, e’: E?

   undefined

9. What is the value of rest (rest (insert (new, e))) for any e: E, assuming that LinearContainer has access to Stack?

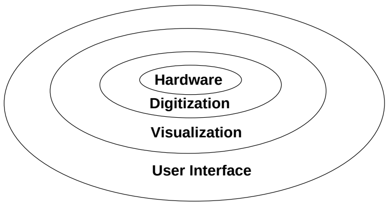
   undefined

10. What is the value of first (insert (rest (insert (new, e)), e')) for any e, e': E, assuming that LinearContainer has access to Stack?

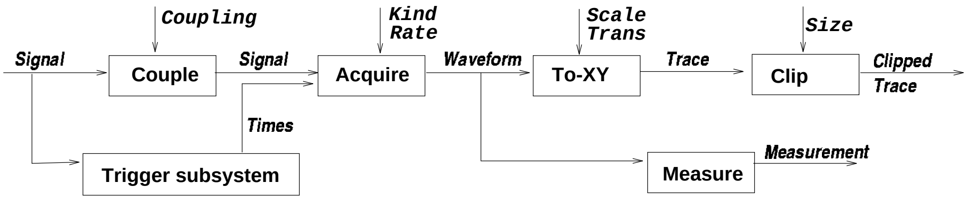
e'

5. [20 marks]

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



Architecture 1



Architecture 2

1. [8 marks] Describe in relation to *Architecture 2* one major problem with *Architecture 1.*

   - boundaries of abstraction conflicted with the needs for function interactions e.g., requires all user interactions with an oscilloscope thru the visual representations

   but, users need to affect the functions in all layers:

   e.g., setting channels in Hardware layer; choosing acqusition mode & parameters (e.g., threshold value) in Digitization layer; creating derived waveforms (e.g., scaling factor) in Visualization layer.

2. [8 marks] What kind of mechanism would be needed to display Measurement on the screen?

   A mechanism to:

   - convert raw measurements to scaled, translated, and clipped measurements (font size, etc. need to be determined as well)

3. [4 marks] Pictorially depict a 2-layer architecture which can be obtained from *Architecture 2.*