The University of Texas at Dallas
Department of Computer Science

Test 1 June 20, 2005

Conditions: Closed book Duration: 90 minutes

State assumptions, if there is any

Please write legibly; unreadable answers are NOT answers!

Name: _____

{Please underline last name}

Student Number: _____

1. _____ /20

2. _____ /20

3. _____ /30

4. _____ /20

5. _____ /10

Total _____ /100

1. [20 marks] **page 2/9**

For each of the following ten statements, indicate whether it is true (mark T) or false (mark F).

__T__ This is Test 1 for CS6362.

___ 1. Given n as the number of non-functional requirements and f as the number of functional requirements, the total number of architectural solutions is $O(f * n)$.

___ 2. The quality of an object-oriented system is often determined by the particular design decisions made during the architectural design process.

___ 3. The completeness and consistency of a software architectural description are necessary, but not sufficient, conditions for ensuring that the corresponding implementation will have good quality.

___ 4. A software architecture with high coupling should help achieve efficient and effective integration testing.

___ 5. The mapping between an architectural design and a detailed design corresponding to the architectural design is 1–1.

___ 6. The key concept in MILs is object inheritance.

___ 7. Various surveys indicate that the various notions of software architecture have indeed been used precisely and explicitly by software practitioners in the past.

___ 8. Java is a good language to specify an implicit invocation style of software architecture, since Java's inheritance mechanism nicely captures the semantics of implicit invocation.

___ 9. In software architectural design, the precise specification of architectural constraints has been a bottleneck in achieving a higher level of distributed processing.

___ 10. For a safety-critical system, such as a space shuttle launching system, its architecture should guarantee absolutely that it satisfies all the safety requirements as well as any other important reliability requirements on the system.

2. [20 marks] page 3/9

Circle the best answer to each of the following questions.

2.1 Which is the *most* relevant to software architecture?

1. structured programming
2. dynamic linking library (DLL)
3. quick sort
4. binary search
5. client and server

2.2 Which is the *most* relevant to programming-in-the-small?

1. emphasis in algorithm design
2. communication and coordination difficulties
3. changing requirements
4. system architecture
5. clear problem

2.3 Which is the *most* relevant to an architecture in a shared data style?

1. information hiding
2. even-driven
3. good space performance
4. implicit invocation
5. highly reusable

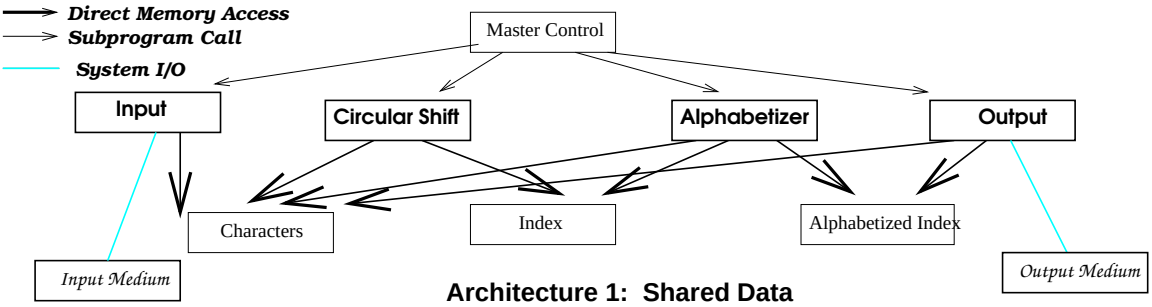
2.4 Which is the *least* relevant to the architecture of a house?

1. front view
2. components
3. connections
4. layout of electrical wires
5. design decisions

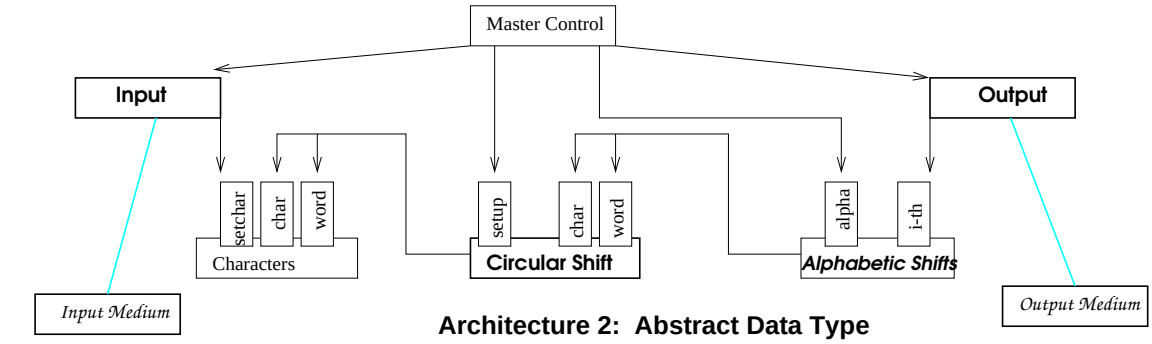
2.5 Which is the *least* relevant to the types of software architectural components?

1. control
2. data
3. pointer variable
4. process

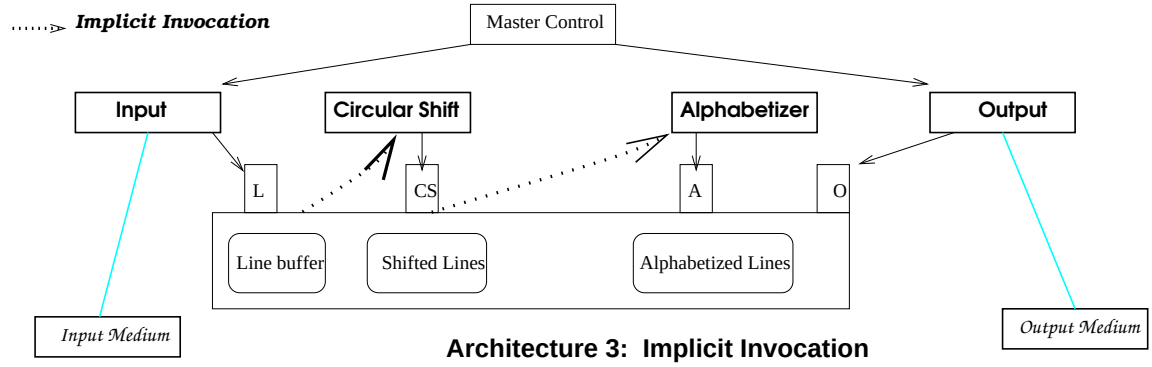
Consider the following four architectures for the KWIC problem.



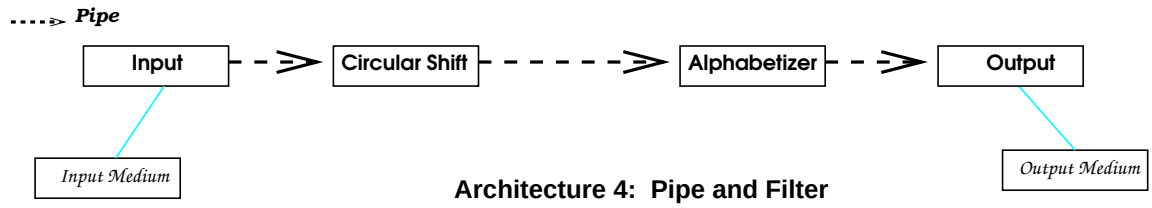
Architecture 1: Shared Data



Architecture 2: Abstract Data Type



Architecture 3: Implicit Invocation



Architecture 4: Pipe and Filter

For the following questions, the following two titles are the input strings (separated by a $ sign):

easy to use$secure system

3.1 Consider *Architecture 1.* As discussed in class, **Circular Shift** produces, for each line of circular shift, the starting index of the source line and the offset from the starting position.

Write down the indices for the first (1) and third (3) circular shifts.

3.2 Consider *Architecture 1.* Assume that a noise eliminator module has been added between **Circular Shift** and **Alphabetizer**. As discussed in class, **Alphabetizer** converts "Index" to an "Alphabetized Index" by listing the circular shifts alphabetically, here assuming that the input comes from the noise eiliminator module.

Write down the indices for the first (1) and third (3) alphabetized circular shifts.

3.3 Consider *Architecture 2.* What would *char (2, 2, 2)* of **Characters** return?

3.4 Again consider *Architecture 2.* What would *char (2, 2, 2)* of **Circular Shift** return?

3.5 Consider *Architecture 3* How would you improve the architecture? Explain your answer.

3.6 Consider *Architecture 4*. Formulate (i.e., mathematical formulae) the performance of the system *in terms of the components and connections* of the system, assuming that there is no concurrency in the system.

Consider the following declaration of an ADT:

```
Ship (P, C): trait /* P (e.g., ``Rose" and ``Jack") are elements of C */
      introduces
            isIn: P, C -> Bool
            out: P, C -> C
            in: P, C -> C
            new: -> C
      asserts
            C generated by new, in

            forall p: P, p': P, c: C
                  out (p, new) == new

                  out (p', in (p, c)) == out (p, c)           if (p == p')
                  out (p', in (p, c)) == in (p, out (p', c))  otherwise

                  ~ isIn (p, new)

                  isIn (p', in (p, c))               if (p == p')
                  isIn (p', in (p, c)) == isIn (p', c) otherwise

      implies
            Airplane (new for start, in for on, out for off, isIn for fly)
```

Now, suppose you are using a theorem prover, called *TP*. For each of the following questions, show your proof. (Each correct answer is worth 2 point; each correct proof is worth 2 points).

4.1 What should TP return as the value of isIn ("Jack", in ("jack", new ("Rose")))?

4.2 What should TP return as the value of isIn ("Rose", out ("Jack", new))?

4.3 What should TP return as the value of isIn ("Jack", in ("Rose", out ("Rose", in ("Jack", new))))?

4.4 What should TP return as the value of fly ("Jack", off ("Rose", on ("Jack", start)))?

4.5 How is this specification better, with respect to a specification in MILs?

5. [10 marks] page 9/9

Consider the following declarations:

```
module M1
       provides: v, c;
       requires: d;
       consist-of: module M12, function F11

       module M12
              provides: c;
              requires: d;
              has-access-to: module M2
              int c, boolean d
       end M12

       function F11
              provides: v;
              requires: c;
              boolean v, int c
       end F11
end M1

module M2
       provides: d,e;
       int d, char e
end M2
```

2.1 [3 marks] List the set of variables that M12 can access.

2.2 [2 marks] The specification above has one inconsistency. What is the inconsistency?

2.3 [5 marks] Suppose M has access to L. Also suppose L' is a child of L. Precisely specify a constraint that M has access to L' through L. There should not be a direct has-access-to relation from M to L' in the specification. (Hint: Use a set notation)