The University of Texas at Dallas
Department of Computer Science

Test 2 July 18, 2005

Conditions: Closed book Duration: 90 minutes

Write legibly; unreadable answers are not answers!

Each answer should be concise and precise! Indicate any assumptions

Name: _____

{Please underline last name}

Student Number: _____

1. _____ /15

2. _____ /15

3. _____ /25

4. _____ /20

5. _____ /25

Total _____ /100

1. [15 marks] **page 2/9**

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 second test for CS6362.

___ 1. In a layered architecture, a lower layer usually implements the services that its upper layer needs.

___ 2. Change in an architectural specification does not necessarily imply change in the corresponding functional requirements specification.

___ 3. The quality of an implementation depends entirely by the degree to which the implementation preserves the structure (i.e., components and interactions between the components) of its corresponding architecture.

___ 4. Use of JavaBeans enhances low cohesion and composability.

___ 5. Use of Java applets leads to thin clients.

2. [15 marks] **page 3/9**

Circle the best answer to each of the following questions.

1. Which is the *most* likely reason a blackboard architecture is chosen over a pipe-and-filter architecture?

1. concurrency
2. process independence
3. events
4. performance predictability
5. simplicity

2. Which is the *least* likely constituent of a software architecture?

1. communication protocol
2. data structure
3. trigger condition
4. daemon
5. timing service

3. Which is the *most* relevant to integration testing?

1. tradeoff analysis
2. design patterns
3. smart agents
4. visibility of the server representation
5. consistency among JavaBeans interfaces

4. Which is the *least* relevant to *J2EE*?

1. architecture
2. EJB
3. absolute security
4. component-based software engineering
5. layered

5. Which is the *most* relevant to design pattern?

1. Z
2. workflow
3. blackboard system
4. MVC
5. MILs

3. [25 marks]

3.1. (10 points) What is the key issue with an architecture in which every component communicates with every other component.

3.2. (15 points) Sue and Sam are sitting in a classroom, where Tim is the instructor. Now the class proceeds as Tim asks questions on a particular topic, and the students answer them. Tim then leads the discussion by asking deeper questions in response to the answers from the students, in an incremental manner. During the discussion, the students do not talk with each other, while Tim writes down both questions and answers on the blackboard.

3.2.1 (5 points) List who (or what) plays the role of knowledge source(s), blackboard and control.

3.2.2 (10 points) Now suppose both Sue and Sam can also talk to each other, while at the same time Tim no longer uses any blackboard. What would be the architectural style of the discussion in this case? What would be the weaknesses associated with this new style?

4. [20 marks]

Suppose you are to build a system to help buyers and sellers, in a real estate market, who are interested only in the price and location of the real estate.

4.1. (10 points) Depict a diagram of a software architecture in an implicit invocation style for the system. You should show all the needed interfaces as precisely as possible.

4.2. (5 points) Depict a UML class diagram showing how the architecture in **5.1** can be implemented using the Java Event Model.

4.3. (5 points) Depict a sequence diagram for the Observer design pattern.

Consider the following definition:

```
Voltage == Z X {Volt}            /* "X" refers to a cross product */
Time == N X {Second}

Waveform == Time -|-> Voltage    /* "-|->" refers to a partial function */

Segment == {w: Waveform | dom w subsetOf ContigPeriod},
           where ContigPeriod == {st: P1 Time| (exists t1, t2: Time .
                                    st = {t: Time| t1<t<t2})}
Coord == R X {Metre}
Point == Coord X Coord
Trace == Time -|-> Point


___ Scale ____________________________________________________________
| segment: Segment
| HScale: Time
| VScale: Voltage
| scaled: Trace
----------------------------------------------------------------------
| scaled =   (lambda t: dom seqment .
|                     ( (t - min(dom segment)) / HScale,
|                       (segment(t) - min(ran segment)) / VScale) )
|                       /* "dom" refers to domain, "ran" refers to range */
----------------------------------------------------------------------


___ Translate ________________________________________________
| scaled: Trace
| HOffset, VOffset: Coord
| moved: Trace
---------------------------------------------------------------
| moved = (lambda t: dom scaled . (second(scaled(t)) + HOffset,
|                                  first(scaled(t)) - VOffset))
---------------------------------------------------------------


__ Clip ______________________________________________________
| moved: Trace
| clipped: Trace
---------------------------------------------------------------
| let screen == {(x, y): Point| HMin < x < 7 and VMin < y < 8} .
|               clipped = moved |> screen   /* "|>" is the range restrictor
---------------------------------------------------------------
```

In the following, measurement units will be omitted. Also note that "|–" denotes mapping at the instance level. Assume

```
segment = {4  |-  2, 5  |-  10, 6  |-  5, 7  |-  1, 8  |-  11}
HScale = VScale = 1
HOffset = 1
VOffset = 1
HMin = 1
VMin = 2
```

5.1. Compute the value of "scaled".

5.2. Compute the value of "moved" corresponding to the above.

5.3 Compute the value of "clipped".

5.4 (10 points) Define a schema "Flipped" which can be used to display the mirror image of "clipped" along the horizontal axis.