1. (15 pts.) Match each description below to the SINGLE MOST APPROPRIATE ARCHITECTURAL PATTERN OR TYPE among the following. (Note: patterns or types may apply to none, one, or more than one description.)

A. Model-View-Controller
B. Repository
C. Data-flow architecture
D. Transaction Processing
E. Abstract machine model
F. Client-server
G. Event-based
H. Data Processing
I. P2P
J. Language Processing
K. Centralized
L. Distributed component

___ Data driven applications that process data in batches without explicit user intervention during the processing.

___ System reacts to external stimuli for organizing control

___ When transformations are sequential, this pattern is known as a “batch sequential model.”

___ Very commonly used run-time organization for distributed systems connected using Internet protocols.

___ Examples include compilers and command interpreters.

___ When a layer interface changes, only the adjacent layer is affected, but it is often difficult / artificial to structure systems in this way.

___ Shows how data and processing are distributed across a range of processors.

___ May include an interpreter to execute instructions in the language being processed.

(cont’d on next page)
1. (cont’d)

___ The processing of the data in a system is organized so that each processing component is discrete and carries out one type of data transformation. The data flows from one component to another for processing.

___ Used when you have a system in which large volumes of information are generated that has to be stored for a long time. All data can be managed consistently (e.g., backups done at the same time) as it is all in one place.

___ Used when there are multiple ways to view and interact with data, or when the future requirements for interaction and presentation of data are unknown.

___ Systems process user requests for information from a database or to update the database. Examples include E-commerce and reservation systems.

___ One component calls the other components.

___ Example would be:

![Dexter Reference Model](image-url)

___ Example would be:
2. (4 pts.) What are reference architectures? What is the relationship between a reference architectures and a generic application architectures in some domain?

3. Sommerville notes that there are differing views about whether or not software architects should use the UML for architectural descriptions. He refers to a survey by Lange, et al., from 2006 suggesting that when UML was used, it was mostly applied in a “loose and informal way.”

   a. (2 pts.) What was the view of Lange, et al., about this?

   b. (4 pts.) What is Sommerville’s view about this and what reason(s) does he give?

4. (3 pts.) The “information-hiding based approach” was one of several heuristics described in class for identifying objects/object classes during object-oriented design. It is based on a very general definition of information hiding that is often attributed to David Parnas. (In particular, the concept extends beyond the scope of object-oriented systems.) What is Parnas’ definition of information hiding?

5. (4 pts.) In describing host-target development, Sommerville mentions UML deployment diagrams. What information is shown in a deployment diagram? Be specific.
6. (3 pts.) Which one of the following best describes the role of “Sequence diagrams” when used to model an object-oriented design? (Circle ONE only.)

   a. They show how the system as a unit interacts with its environment as it is used.
   b. They show how the design is organized into logically related groups of objects. (The actual organization of objects in the system as implemented may be different.)
   c. They are primarily used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves.
   d. They are used to identify the other systems in the environment of the system being developed.
   e. They show how an object responds to different service requests and the internal transitions that are triggered by these requests.
   f. They show the hardware and software in the system and the middleware used to connect the different components in the system.

7. (3 pts.) Sommerville identifies several costs of reusing existing components or systems. Which one of the following is NOT explicitly identified as a cost of reuse? (Circle one only.)

   a. The time spent in looking for software to reuse and assessing whether or not it meets the needs.
   b. Where applicable, the costs of buying the reusable software.
   c. The costs of adapting and configuring reusable software components or systems to reflect the requirements of the system being developing.
   d. The costs of hardware maintenance and support software licenses required for the reusable software.
   e. None of the above – all were explicitly identified.

8. a. (3 pts.) Sommerville describes three fundamental configuration management activities. Two of the three activities are version management and system integration. What is the third activity and what function(s) does it support?

   b. (2 pts.) Which of the three activities help developers define what versions of components are used to create each version of a system.
9. (12 pts.) Match each description below to the **SINGLE MOST APPROPRIATE TERM** among the following. (Note: terms may apply to none, one, or more than one description.)

<table>
<thead>
<tr>
<th>Description</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Validation testing</td>
<td>I. Black-box testing</td>
</tr>
<tr>
<td>B. Defect testing</td>
<td>J. White-box testing</td>
</tr>
<tr>
<td>C. Release testing</td>
<td>K. Operational profile</td>
</tr>
<tr>
<td>D. Alpha testing</td>
<td>L. Software inspections</td>
</tr>
<tr>
<td>E. Performance testing</td>
<td>M. Beta testing</td>
</tr>
<tr>
<td>F. Stress testing</td>
<td>O. Security test</td>
</tr>
<tr>
<td>G. Partition testing</td>
<td>P. “Soak” test</td>
</tr>
<tr>
<td>H. Regression testing</td>
<td>R. MVQ</td>
</tr>
</tbody>
</table>

___ In contrast to system testing, the objective is to check that the system is good enough for external use; systematic forms include “requirements-based testing” and “scenario testing.”

___ End-user testing performed in the customer environment prior to general release.

___ Undertaken to *demonstrate* that a system operates as intended.

___ Involve people examining a system artifact (requirements, design documents, source code, etc.), usually with the aim of discovering anomalies and defects.

___ Re-running of one or more test cases, after some program change, that ran without revealing faults prior to the change.

___ V&V activity that can consider quality attributes such as compliance with standards, portability, and maintainability.

___ Undertaken to expose situations in which the behavior of the software is incorrect, undesirable, or does NOT conform to its specification.

___ Intended to reflect the expected usage of a system in some environment.

___ May be automated by combining a keystroke recorder and playback tool with a data/output comparator.

___ Focus is on typical requirements that systems exhibit “graceful” failures when overloaded.

___ Requires being comfortable with testing less and knowingly shipping buggier software faster.

___ Choosing tests from identified groups of inputs that have common attributes and are therefore expected to be processed by a program in the same way.

10. (4 pts.) Rajlich and Bennett (2000) identify four general software life-cycle phases: *initial development, phase-out, servicing, and evolution*. Briefly explain the differences between *servicing* and *evolution* in terms of the nature of changes systems may be subject to in each.
11. (4 pts.) Sommerville describes *path testing* as a testing strategy that aims to exercise every (feasible) independent execution path through a component or program. “If every independent path is executed, then all statements in the component must have been executed at least once. All conditional statement are tested for both true and false cases.” Which one of the following best describes the relationship between *path testing* (as defined above) and *exhaustive testing*? (Circle one only.)

a. *Path testing* subsumes *exhaustive testing*. (I.e., if you have exercised every feasible independent path in a program with test cases, then you will have tested the program exhaustively.)

b. Exhaustive testing subsumes path testing. (I.e., if you have tested a program exhaustively, then you will have exercised every feasible independent path in a program with test cases.)

c. Both (a) and (b) – i.e., the two testing criteria subsume each other; they are equivalent.

d. Neither (a) nor (b) – i.e., neither testing criteria is subsumed by the other; they are independent.

e. (none of the above)

12. a. (3 pts.) What point does Sommerville make with the figure below concerning how one should think about the relationship between initial software development and software evolution?

b. (3 pts.) Sommerville also notes that there is a critical difference in the first stage of change implementation during software evolution that is not explicitly represented in this figure. What is the difference and under what circumstances is it most likely to be critical?
13. a. (10 pts.) Provide a graphical model of the Test-Driven Development (TDD) process as it would be employed in support of agile methods such as Extreme Programming. Your model should incorporate only the following process activities: (listed in no particular order)

   A: “Run the test together with other tests that have already been implemented.”
   (show two possible outcomes of this activity in your model: “pass” and “fail”)
   B: “Implement the new functionality and refactor.”
   C: “Write and implement a test for the new functionality.”
   D: “Identify the new functionality.”

   Use ovals (labeled A, B, C, or D to represent the four activities identified above), arrows, diamonds, etc., to model the TDD process as discussed in class. (An illustrative example of a process model is shown below.)

   Example process model:

   ![Test-Driven Development process model]

   **Test-Driven Development** process model:

b. (3 pts.) Briefly explain under what circumstances it would be possible for a programmer to correctly cycle through this process for a new (i.e., not-yet-implemented) increment of functionality without modifying the program in any way?
14. (8 pts.) Consider the following statements related to program evolution dynamics. Circle either "true" or "false" as (most) appropriate. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the observation.

a. Lehman and Belady claim that their laws are likely to be true for all types of large organizational software systems (what they call E-type systems). true false

b. According to Sommerville, the “Law of Organizational Stability,” which suggests that most large programming projects work in what he terms a saturated state, is perhaps the most contentious of Lehman’s laws. true false

c. Lehman’s “Law of Continuing Change” implies that systems that do not change over time are either not used in a real-world environment, or are becoming progressively less useful. true false

d. The “Law of Large program evolution” suggests that large systems have a dynamic of their own that is established at an early stage in the development process. true false

15. (3 pts.) Which one of the following best reflects the point Sommerville makes in connection with how “contractual responsibility” can affect maintenance costs? (Circle ONE only.)

a. Maintenance costs can be dramatically reduced if development contracts stipulate that development teams NOT be broken-up and people assigned to new projects after a system is delivered. true false

b. Unless prior training of maintenance personnel is contractually mandated, a lot of the effort during the maintenance process is taken up with understanding the existing systems before changes can be implemented. true false

c. If the maintenance contract is given to a company other than the original system developer, there may be little or no incentive for a development team to write the software so that it is easy to change. true false

d. If the contract to maintain a system is part of the system development contract, a development team may cut corners to save effort during development since its members understand the system and the background of system design decisions. true false

e. Contractual requirements related to the use of modern software engineering techniques, system structure, system documentation, and configuration management can significantly impact maintenance costs. true false

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

____________________
SIGNATURE