1. Sommerville notes that the distinction between generic and customized (bespoke) software product types is becoming increasingly blurred.
   a. (2 pts.) Provide brief definitions of generic and customized software products that make the distinction between the two types clear.
   b. (3 pts.) What is his explanation for why the distinction is becoming increasingly blurred?

2. (3 pts.) Sommerville argues that the systematic and organized approach underlying software engineering (as opposed to just writing a program as if it was a personal programming project) is important for two reasons. One of these is that we need to be able to produce reliable and trustworthy systems economically and quickly. Briefly describe the OTHER reason he cites.
3. (3 pts.) Sommerville notes that in any situation where different people have different views and objectives, software engineers are likely to be faced with ethical dilemmas. One example he gives concerns a company that is responsible for developing a safety critical system and, because of time pressure, falsifies safety validation records. What does Sommerville advise that a software engineer employed by such a company do in this situation? (Circle ONE only.)

a. Since there are no absolutes when it comes to safety, he feels an engineer in this situation must make up his own mind about what to do, taking into account the potential for damage, the extent of the damage, and the people affected by the damage.

b. Since the engineer is an employee of the company, his responsibility is to maintain confidentiality and resolve the situation in such a way that protects his employer.

c. It is the engineer’s responsibility to alert the customer or to publicize, in some way, that the delivered system may be unsafe.

d. As suggested by Laudon (1995) and Johnson (2001), the engineer should consider the situation from a philosophical standpoint where the basic principles of ethics are considered, and then decide how to act accordingly.

e. (None of the above.)

4. (3 pts.) Sommerville notes that while there is no “ideal” software process, there is scope for improving the software process in many organizations. Aside from replacing outdated techniques and taking advantage of the best practice in industrial software engineering, which one of the following general actions does he suggest an organization can take to improve its software processes? (Circle ONE only.)

a. Use a more structured, plan driven-process.

b. Use a less formal, agile process that is easier to change in response to changing customer requirements.

c. Increase diversity in software processes across the organization in order to take advantage of the benefits different processes can provide.

d. Decrease diversity in software processes across the organization through process standardization.

e. Use a modern generic process that is organized into phases but separates activities from these phases.

f. (None of the above.)
5. (10 pts.) Match each description / image below to the **SINGLE MOST APPROPRIATE TERM** among the following. (Note: terms may apply to none, one, or more than one description / image.)

A. Boehm’s Spiral Development
B. Throw-away Prototyping
C. Simulation
D. Reuse-based Development
E. Volere
F. RUP
G. Incremental Development
H. Cleanroom software engineering
I. Waterfall
J. COCOMO

___ An example of a formal development process, originally developed by IBM. Software correctness is demonstrated using a formal approach; there is no unit testing for defects in the process.

___ May not be compatible with large organizations where bureaucratic procedures have evolved over time, for example, to ensure that software properly implements external regulations.

___ Because producing and approving documents can be costly, it is normal to freeze parts of the development, such as specification, and to continue with the later development stages. Problems are left for later resolution, ignored, or programmed around.

___ Reflects the way that we solve problems – moving toward a solution in a series of steps, backtracking when we realize that we have made a mistake.

___ In addition to inevitable requirements compromises, some control over system evolution is lost as new versions of some components are not under the control of the organization using them.

___ Combines change avoidance with change tolerance; it assumes that changes are a result of things that can go wrong in a project, and includes explicit management activities to reduce the likelihood and impact of such events.

___ A phased model that supports iteration in two ways: each phase may be enacted in an iterative way with the results developed incrementally AND the whole set of phases may also be enacted incrementally.

___ A practice perspective describes six fundamental best software engineering practices that are recommended for use in systems development.

___ Process model:

---

___ Process model:
6. (7 pts.) Match each description / image below to the **SINGLE MOST APPROPRIATE TERM** among the following. (Note: terms may apply to none, one, or more than one description / image.)

- A. mockup
- B. breadboard
- C. horizontal prototyping
- D. vertical prototyping
- E. throw-away prototyping
- F. evolutionary prototyping
- G. experimental prototyping
- H. “Wizard of Oz” prototyping
- I. simulation
- J. compound documents
- K. component and application assembly
- L. LISP, APL, SmallTalk
- M. RAD environments
- N. paper prototyping

___ Objective is to evaluate a proposed solution for feasibility / performance.

___ “To realistically evaluate the dynamic behavior of different interface designs for a new chess playing program before implementing its core (chess playing) engine, Company XYZ hired a chess master to simulate the engine in real-time from another city as test subjects played chess with prototypes reflecting the different designs.”

___ “A non-functioning but realistic looking prototype was developed to test the market appeal of the new devise.”

___ “It was decided to develop a prototype that would very realistically reflect only that system functionality that handles emergency operator over-ride situations.”

___ Represented by:

___ Could make use of:

___ May result in:
7. (3 pts.) Which one of the following best describes a requirement’s *rationale*? (Circle ONE only.)

   a. A description of the intention of the requirement.
   b. A description of the criteria by which the correctness of the system with respect to the requirement would be determined.
   c. A description of the extent to which procurers or end-users want or need the requirement to be implemented.
   d. A description of political or organizational factors that may affect the requirement.
   e. A description of the reason for the requirement.

8. Sommerville notes that in practice, the principles underlying agile methods can sometimes be difficult to realize. For example, writing a contract that includes a software requirements document is usually impossible since incremental specification is inherent in agile methods.

   a. (2 pts.) Since agile methods generally cannot utilize contracts in which the customer pays for the development of a specific set of requirements, what arrangement for payment is normally specified in contracts when agile methods are used?

   b. (2 pts.) What specific risk does Sommerville describe in connection with this arrangement?

9. a. (2 pts.) Sommerville notes that interviews undertaken with system stakeholders during the requirements engineering process may be of two types: *closed* and *open*. Briefly explain the difference between closed and open interviews.

   b. (3 pts.) Sommerville argues that interviews are NOT an effective technique for eliciting knowledge about organizational requirements and constraints. Briefly explain why.
10. (6 pts.) Some agile methods enthusiasts argue that it is a waste of time to write formal documentation that describes a system in order to make it easier for people changing the system to understand it. Instead, they claim that the key to implementing maintainable software is to produce high-quality, readable code. Agile practices therefore emphasize the importance of writing well-structured code and investing effort in code improvement. Thus, (the argument goes) the lack of documentation should not be a problem in maintaining systems developed using an agile approach. What is Sommerville’s thinking about this argument? (Hint: based on personal experience, he feels that one document, in particular, is critical to system maintenance. (1) What document is it, (2) why is it critical, and (3) to what extent do agile methods typically support this need?) Be specific.

11. The following FALSE statements concerning the “Scrum” project management approach appeared in a “Circle either ‘true’ of ‘false’…” exam problem in fall, 2010. In the space provided below each statement, briefly correct the statement as indicated by the provided hint.

   a. (4 pts.) The role of the Scrum Master is to determine what should be delivered, when it should be delivered, and who will work on the deliverables. (Hint: briefly describe what the role of a Scrum Master actually is.)

   b. (2 pts.) Sprints are variable length (normally 2-4 weeks) and correspond to the phases of development in plan-based development. (Hint: both of the claims in this statement are untrue. Re-write the statement correcting the two mistakes.)

   c. (2 pts.) At the end of a sprint, the development work is reviewed and presented to the Scrum Master. The next sprint cycle then begins. (Hint: the work done is NOT presented to the Scrum Master. To whom IS it presented?)
12. (10 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.)

- A. system goal
- B. user requirements
- C. system constraint
- D. system requirements
- E. domain requirement
- F. system attribute
- G. external requirement
- H. process constraint
- I. operational specification
- J. form/template-based specification
- K. interface specification
- L. scenario
- M. Software Requirements Document (a.k.a. an “SRS”)

___ Example would be: "system response time under full operational load"

___ Usually suffers from not being verifiable.

___ Example would be: "Use of the Eclipse IDE is mandatory."

___ Example would be: "The system is required to meet CENELEC standards EN 50126, EN 50128, and EN 5019 before being approved for deployment."

___ Example would be: "At least two fully functioning thrusters are required to reach escape velocity at 14.5038 psi."

___ Example would be: "Should integrate easily with customers’ other systems."

___ Official statement of what is required of system developers; it should include both user and system requirements.

___ Example would be: “probability of system unavailability must be ≤ .05”

___ Example would be:

```java
Service Utility {
    package util
    @sendNotification(notificationMessage : string) : void >> msg
    @getId():long >> admin
    @saveLogEntry(entry:string, type:logEntryType):LogEntry >> log
    @getTimestamp():dateTime >> util
}

// admin
ValueObject AdministrativeStatus {
    package admin
    source:string
    status:string
    subject:string
}

// logging
ValueObject logEntry {
    package log
}

___ Example would be:

**REPEAT**

Select the next (initially, the first) Effect.

Tracing back through the graph (right to left), find all feasible combinations of connected Cause values that result in the Effect being True.

For each new such combination found:

Determine values of all other Effects, and
Enter values for each Cause and Effect in a new column of the test case coverage matrix.

**UNTIL** each Effect has been selected.
13. (7 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 Processing  
H. Information systems

___ Used in data-driven systems where a blackboard model triggers components when particular data becomes available in a shared database.

___ Used when there is a requirement for multi-level security.

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

___ Also known as a “pipe and filter architecture”. The name comes from the original Unix system where it was possible to link processes (“filters”) using “pipes”.

___ Application architecture that allows controlled access to a large base of information, such as a library catalog, a flight timetable, or the records of patients in a hospital.

___ Example would be:

___ Example would be:
14. (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 specific definition of *information hiding* attributed to David Parnas. What is Parnas’ definition of *information hiding*?

15. (3 pts.) Which one of the following best describes the role of “Context Models” when used in object-oriented design? (Circle ONE only.)

a. 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.)

b. They show how an object responds to different service requests and the internal transitions that are triggered by these requests.

c. They are used to identify the *other systems* in the environment of the system being developed.

d. They show the sequence of object interactions associated with system uses. (Time is represented vertically; models are read top to bottom.)

e. They show how the system interacts dynamically with its environment as it is used.

f. They show the hardware and software in the system and the middleware used to connect the different components in the system.

16. (3 pts.) Sommerville describes three major configuration management activities. Two of these are problem tracking (keeping track of reported bugs, who is working on them, and when they are fixed) and system integration (used to build a system automatically by compiling and linking the required components). Describe the third major configuration management activity identified by Sommerville.

17. (2 pts.) Sommerville describes three open source software licensing models, one of which is the GNU General Public License (GPL), a so-called “reciprocal” license. Briefly describe the obligations (if any) of users who utilize open source software under this license.
18. (14 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>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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. Test-driven development</td>
</tr>
<tr>
<td>H. Regression testing</td>
<td>R. Exhaustive testing</td>
</tr>
</tbody>
</table>

___ 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.

___ 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.

___ Would **guarantee the program is error-free** if no defects were found.

___ Involves implementing new program functionality **AFTER** implementing and running a test for that functionality.

___ 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.

___ End-user testing performed at the developer’s site.

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

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

___ Usually involves a series of tests where the load on a system is steadily increased.
19. (4 pts.) In his introduction to Chapter 9 on “Software evolution,” Sommerville mentions “brownfield software development” (a term coined by Hopkins and Jenkins) in connection with an important point he makes about difficulties and costs of software evolution. What was the specific point he was making in this context?

20. Sommerville describes Lehman’s third “law” concerning system change, the law of “Large program evolution,” as being, perhaps, “the most interesting and the most contentious.”

a. (3 pts.) Briefly state (in your own words, if you like) what the law says.

b. (8 pts.) Belady and Lehman suggest that this law is a consequence of two types of factors. Identify these and briefly describe (1) their nature/origin and (2) how they are related to the law. Be specific.