1. (3 pts.) Sommerville argues that while software engineering is often criticized as being inadequate for modern software development, many so-called software failures are actually a consequence of two (other) factors. One of these is the fact that many companies that do not produce software in their everyday work have “drifted” into software development as their products and services have evolved. Consequently, their software is often more expensive and less reliable than it should be. Briefly describe the other factor he cites.

2. (3 pts.) What, according to Sommerville, is the single “best” software engineering development method or process in use today? (Circle ONE only.)

   a. the new rational waterfall process
   b. spiral development
   c. plan-driven development
   d. agile development
   e. incremental development
   f. incremental delivery
   g. reuse-based development
   h. the Rational Unified Process (RUP)
   i. (none of the above)
3. (3 pts.) In describing Richard Fairley’s observations about the intangibility of software from his keynote address at the 8th Conference on Software Engineering Education, a reference was made to the (probably apocryphal) story of a Boeing aeronautical engineer asking a software engineer about how much the software being installed on one of the first aircraft equipped with computers weighed. How, according to the story, did the software engineer respond to this question? (Circle ONE only.)

a. By suggesting that the weight of the software would depend on “the state of magnetization, voltage level, and current flow” in the on-board computer.

b. By placing a short strip of magnetic tape on the palm of the aeronautical engineer’s hand and responding, “about that much.”

c. By pointing to the donut he was about to eat and saying, “about as much as the hole in my donut weighs.”

d. By responding, “about as much as your beard weighs.” (The aeronautical engineer was beardless.)

e. By showing the aeronautical engineer a “deck” of punched cards and pointing out that the weight of the software was that of the holes in the cards.

4. (3 pts.) One of the issues of professional and ethical responsibility for software engineers discussed in class was that of competence. Which ONE of the following best describes the issue as discussed?

a. You should normally respect the competence of your employers or clients irrespective of whether a formal competence agreement has been signed.

b. You should be aware of local laws governing the required competence of intellectual property such as patents and copyright. You should be careful to ensure that the competence of employers and clients is protected.

c. Software engineers should strive to remain competent in all areas of software engineering if they are to be respected as professionals.

d. Software engineers should always act in a competent manner that is consistent with the public interest.

e. Software engineers should not misrepresent their level of competence or knowingly accept work which is outside their area of competence.

5. (2 pts.) Fred Brooks, author of The Mythical Man-Month, described the relative difficulty and importance of one software engineering activity in these terms: “No other part of the conceptual work is as difficult... No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.” What activity was he describing?
6. (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. Boehm’s Spiral Development
- B. Throw-away Prototyping
- C. Simulation
- D. Reuse-based Development
- E. Volere
- F. RUP
- G. Incremental Development
- H. Extreme Programming
- I. Waterfall
- J. COCOMO

___ Often described as a “deliverable oriented development model”; deliverables are intended to provide good process visibility.

___ Specification, development, and validation are interleaved; may be plan-driven or agile.

___ Normally described from 3 perspectives: dynamic (showing phases over time), static (showing process activities), and practice (suggesting good practice).

___ The system is assembled from existing web services, components, or COTS systems; may be plan-driven or agile.

___ Utilizes "test-first development" – writing tests from scenarios before coding to clarify requirements.

___ Plan-driven model mostly used for large systems engineering projects where a system is developed at several sites.

___ Model with discrete phases related more closely to business rather than technical concerns.

___ Process model often credited with introducing the risk-driven approach to development.

___ Focus is on people – not process – through pair programming, collective ownership, and a process that avoids long working hours.

___ A “structured method” to elicit, analyze, and document requirements.

7. (3 pts.) Which one of the following best describes what it means to say that a requirement is **verifiable**? (Circle ONE only.)

a. The requirement can be changed with little or no effect on other system requirements.

b. Correctness of the system with respect to the requirement can be determined.

c. Procurers or end-users of the system are able to understand the requirement.

d. Consistency of the requirement with user needs/desires can be determined.

e. The origin of the requirement is clearly documented.
8. (3 pts.) Which one of the following is generally NOT considered to be a characteristic of the **waterfall model**? (Circle ONE only.)

a. The result of each phase is one or more documents that are approved (“signed-off”).

b. The principal stages of the model map onto fundamental development activities.

c. The software specification, design, and implementation are broken down into a series of increments that are each developed in turn.

d. A phase of development should not start until the previous phase has finished.

e. None of the above – ALL are characteristics of the **waterfall model**.

9. Sommerville notes that to decide on the appropriate balance between a plan-based and an agile development approach, one needs to answer a range of technical, human, and organization related questions. Two such questions appear below. Briefly describe their relevance to this issue as explained by Sommerville.

a. (2 pts.) **How good are the designers and programmers in the development team?**

b. (2 pts.) **Is the system subject to external regulations?**

10. (4 pts.) Sommerville describes two types of “scaling” challenges for agile methods: “scaling-out” and “scaling-up.” Briefly explain what these terms mean in the specific context of agile methods.
11. (+/-16 pts.) Circle either “true” or “false”, as appropriate, for each of the following statements concerning the "Scrum" project management approach. Note that to compensate for random guessing, you will receive +2 pts. for each correct answer and -2 pts. for each incorrect answer. (No points will be added to or subtracted from your score if you do not circle an answer.)

a. Scrum’s focus is on managing iterative development rather than specific agile practices.  
   true  false

b. 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.  
   true  false

c. “Scrum” is a cricket term for a top-order batsman capable of batting for a long duration throughout the innings.  
   true  false

d. Once the features and functionality to be developed during a sprint are selected, the team is isolated from the customer with all communications channelled through the Scrum master.  
   true  false

e. The role of the Scrum blocker is to protect the development team from external distractions.  
   true  false

f. Sprints are variable length (normally 2-4 weeks) and correspond to the phases of development in plan-based development.  
   true  false

g. The starting point for each sprint is the product backlog, which is the list of work to be done on the project.  
   true  false

h. At the end of a sprint, the development work is reviewed and presented to the Scrum Master. The next sprint cycle then begins.  
   true  false

12. (4 pts.) Sommerville summarizes arguments made by Bass, et al., concerning three specific advantages of explicitly designing and documenting a software architecture. Briefly describe TWO of the three advantages identified.
13. (13 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>A. system goal</td>
<td>H. process constraint</td>
</tr>
<tr>
<td>B. user requirements</td>
<td>I. operational specification</td>
</tr>
<tr>
<td>C. system constraint</td>
<td>J. form/template-based specification</td>
</tr>
<tr>
<td>D. system requirements</td>
<td>K. interface specification</td>
</tr>
<tr>
<td>E. domain requirement</td>
<td>L. scenario</td>
</tr>
<tr>
<td>F. system attribute</td>
<td>M. Software Requirements Document (a.k.a. an “SRS”)</td>
</tr>
<tr>
<td>G. external requirement</td>
<td></td>
</tr>
</tbody>
</table>

___ Example would be: "Must be programmed in Ada."

___ Example would be: "By law, the system must not display patient information to non-clinical hospital personnel."

___ Example would be: "Must be user friendly"

___ Example would be: **pre-condition**: \( N \geq 1 \)  
**post-condition**: there exists an \( i \) in \([1,N]\) such that  
\( \text{BIG} = A[i] \) & for every \( j \) in \([1,N]\), \( \text{BIG} \geq A[j] \)  
& \( A \) is unchanged

___ Example would be: “probability of unavailability \( \leq 0.05\)”

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

___ Example would be: "To avoid boiling, water temperature must be maintained below 100 degrees Celsius."

___ Structured document setting out detailed descriptions of services and constraints precisely.

___ Example would be: "mean time to failure"

___ Example would be a Volere requirement shell.

___ Example would be:

\[
\begin{align*}
\text{t0: The user enters values for input array } A. & \quad \text{The values are } [1, 23, -4, 7, 19]. \\
\text{t1: The user executes program } \text{MAX}. \\
\text{t2: The value of variable } \text{BIG} & \quad \text{is 23 and the values of } A \text{ are } [1, 23, -4, 7, 19].
\end{align*}
\]

___ Usually suffers from not being verifiable.

___ Example would be:

\[
\begin{align*}
\text{BIG} & := A[1] \\
i & := 2 \quad \text{while } i \leq N \text{ do} \\
& \quad \text{if } A[i] > \text{BIG} \text{ then } \text{BIG} := A[i] \text{ end_if} \\
& \quad i := i+1 \\
\text{end}_\text{while}
\end{align*}
\]
14. a. (3 pts.) A research strategy described by Sommerville that is sometimes employed in requirements engineering (RE) is “ethnography.” Briefly explain what ethnography is, how it is conducted, and what useful information it can yield in RE.

b. (3 pts.) What is the difference between “ethnography” and “focused ethnography” and why is the latter potentially more useful that the former in requirements engineering?

15. (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 that is often attributed to David Parnas. Briefly describe the approach as presented in class.

16. (3 pts.) Which of the following best describes the role of “State Diagrams” when used to model an 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 object transitions that are triggered by these requests.

c. They are used to identify other entities (systems, devices, people, etc.) 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 with its environment as it is used.
17. (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 integrating reusable software elements with each other (when using software from different sources) and with any new code being developed.

e. None of the above – all were explicitly identified.

18. (3 pts.) Sommerville identifies three major configuration management activities. Two of these are **version management** (keeping track of the different versions of software components) and **system integration** (used to build a system automatically by compiling and linking the required components). Briefly describe the other major configuration management activity identified by Sommerville.

19. (4 pts.) Sommerville describes three open source software licensing models, one of which is the Berkley Standard Distribution (BSD) License, a so-called “non-reciprocal” agreement. Briefly describe the two main provisions of this model that relate to the types of systems in which BSD code may be included and the obligations (if any) of users who modify BSD code.

20. (4 pts.) Rajlich and Bennett (2000) identify four general software life-cycle phases: *initial development, evolution, servicing, and phase-out*. Briefly describe and contrast the **evolution** and **servicing** phases of their model.
21. (12 pts.) Sommerville notes that the term “testing” can refer to two distinct activities, which he calls **validation testing** and **defect testing**. Briefly compare and contrast these activities in terms of (1) their respective goals, (2) the general nature of test cases employed, and (3) what constitutes a “successful test”.

22. (3 pts.) Fowler, et al., suggest that there are stereotypical situations (called “bad smells”) in which the code of a program can be improved through refactoring. Examples cited include *duplicate code, long methods, switch (case) statements, data clumping, and speculative generality*. Briefly describe what they mean by “speculative generality” and how this situation comes about.

23. (3 pts.) Sommerville observes that change implementation during software evolution can be thought of as an iteration of the development process, where the revisions to the system are designed, implemented, and tested. However, he notes that there is a “critical difference.” What is the “critical difference” he identifies?
24. (+/-8 pts.) Consider the following statements related to program evolution dynamics. Circle either "true" or "false" as (most) appropriate. Note that to compensate for random guessing, you will receive +2 pts. for each correct answer and -2 pts. for each incorrect answer. (No points will be added to or subtracted from your score if you do not circle an answer.)

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 Large program evolution,” which suggests that large systems have a dynamic of their own that is established at an early stage in the development process, is perhaps the most contentious of Lehman’s laws.  
   true  false

c. Lehman’s “Law of Organizational Stability” suggests that most large programming projects work in what he terms a *saturated state*, in which a change to resources or staffing has imperceptible effects on the long term evolution of the system.  
   true  false

d. 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

25. (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. 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.  

b. 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.  

c. 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.  

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.  

e. Contractual requirements related to the use of modern software engineering techniques, system structure, system documentation, and configuration management can significantly impact maintenance costs.
26. (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:

![Process Model](image)

**Test-Driven Development** process model:

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