Sommerville notes that there are two basic kinds of software products: *Generic products* and *Customized (or bespoke) products*.

a. (3 pts.) Which one of the following best describes the distinction between these? (Circle ONE only.)

i. *Generic products* are older, often outdated computer systems or applications, while *customized products* are newer products that may utilize avant-garde technologies.

ii. *Generic products* provide the same functionality as original *customized products*, but are usually marketed using an adopted (nonproprietary) name without advertising.

iii. *Generic products* inherit their traits and functionality from parent systems (i.e., a product line) while *customized products* are independent of any existing application product line.

iv. *Customized products* are based on longstanding application traditions or usage patterns, while *generic products* represent a departure or divergence from application traditions or usage patterns.

v. *Generic products* are produced to be sold on the open market to any customer who is able to buy them, while *customized products* are commissioned by a particular customer.

b. (3 pts.) An important difference between these types of software concerns the software specification. Briefly explain.
2. (4 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 participation in the development of military and nuclear systems. What general advice does Sommerville offer employers and employees who may at some point face this ethical issue? (Circle ONE only.)

a. Since there are no absolutes when it comes to dealing with such issues, he advises employers and employees to decide what to do depending on the potential for human suffering brought about by the use of such systems.

b. He advises employers and employees of an organization to accept any new work assignments, even those associated with military or nuclear systems.

c. As suggested by Laudon (1995) and Johnson (2001), he advises employers and employees to consider such issues from a philosophical standpoint where the basic principles of ethics are considered, and then decide how to act.

d. He advises employers and employees to make their views concerning such issues known to each other in advance (of offering or accepting a position in the organization).

e. He advises employers and employees of an organization to accept any new work assignments, except those associated with military or nuclear weapon systems.

f. (None of the above.)

3. (4 pts.) Consider the reuse-based process model shown below.

![Diagram of reuse-based process model]

Explain why it is essential to have two separate requirements engineering activities in the process. (Please PRINT – do not write cursively. Your writing must be legible in order to receive credit.)
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. Decrease diversity in software processes across the organization through process standardization.

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

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

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

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

f. (None of the above.)

5. (8 pts.) Briefly describe four of the five different uses of prototypes discussed in class. Be specific. (Please PRINT – do not write cursively. Your writing must be legible in order to receive credit.)
6. (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>A. Boehm’s Spiral Development</th>
<th>F. RUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Jackson Systems Development</td>
<td>G. incremental development</td>
</tr>
<tr>
<td>C. Mills’ Incremental Delivery</td>
<td>H. Cleanroom software engineering</td>
</tr>
<tr>
<td>D. Reuse-oriented SE</td>
<td>I. Waterfall model</td>
</tr>
<tr>
<td>E. Volere</td>
<td>J. SADT</td>
</tr>
</tbody>
</table>

___ A hybrid process model derived from work on the UML and the associated Unified Software Development Process.

___ Has the advantage of reducing the amount of software to be developed and so reducing cost and risks.

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

___ A formal development process established at IBM by Harlan Mills, et al. Software correctness is demonstrated using formal methods; there is no unit testing for defects in the process.

___ A phased model where phases are closely related to business rather than technical concerns.

___ Sommerville argues that from a management perspective, there are two problems with this approach: (1) the process is not visible, and (2) the system structure tends to degrade over time.

___ An important innovation is explicitly incorporating the *deployment* of software in a user’s environment as part of the development process.

___ The main difference between this model and other software process models is its explicit recognition of things that can go wrong.

___ Sometimes described as a compromise between the Waterfall model and incremental development.

___ In principle, the result of each phase is one or more documents that are approved. In practice, the process involves feedback from one phase to another. Documents produced in each phase may then have to be modified to reflect the changes made.

___ Some control over system evolution is lost as new versions of some components are not under the control of the organization using them.

___ A fundamental part of agile approaches; it is better than a waterfall approach for most business, e-commerce, and personal systems.

___ Being consistent with other engineering process models, this software process model is still commonly used since it is generally easier to use a common management model for large engineering projects.
7. (10 pts.) Match each description/image/example 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 / 4GL environments  
N. paper prototyping

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

___:

___ Rapid prototype implementation environments supporting database Programming.

___ “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 problem that may arise was referred to by Sommerville as “Pressurizing” the developer.

___ “The complete user interface for a yet-to-be-implemented application was developed using a GUI builder and installed on an android device to test public interest.”

___ A very cost-effective way of obtaining end-user reactions to a draft interface design proposal.

___ “The Turing Test reversed.”

___:

___ Emphasis is on function as opposed to appearance.
8. (14 pts.) Circle either “true” or “false”, as appropriate, for each of the following statements concerning Extreme Programming (XP). 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 statement.

a. For most businesses, rapid development and delivery is rarely the most critical requirement for software systems.  __true__  __false__

b. XP’s focus is on careful project planning, formalized QA, and a controlled, rigorous process.  __true__  __false__

c. Releases of the system are frequent and incrementally add functionality. Enough design is carried out to meet the current requirements and no more.  __true__  __false__

d. An automated unit test framework is used to write tests for a new piece of functionality **before** that functionality itself is implemented.  __true__  __false__

e. Working environments are designed to be comfortable and pleasant since large amounts of overtime are often required immediately prior to system releases.  __true__  __false__

f. Refactoring involves constant improvement in documentation to increase understandability even if there is no immediate need for this.  __true__  __false__

g. Each pair of developers is solely responsible for a different area of the system under development. This helps prevent the formation of “islands of expertise.”  __true__  __false__

9. (3 pts.) Which of the following best describes the role of “Interaction 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.)  __true__  __false__

b. They are used to identify the _other systems_ in the environment of the system being developed.  __true__  __false__

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

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

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

f. They show the hardware and software in the system and the middleware used to connect the different components in the system.  __true__  __false__
10. (15 pts.) It has been suggested that one of the problems of having a user closely involved with a software development team is that they “go native”; that is, they adopt the outlook of the development team and lose sight of the needs of their user colleagues. Suggest three different ways you might avoid this potential problem and briefly justify why each of your suggestions might help in this regard. (Please PRINT – do not write cursively. Your writing must be legible in order to receive credit.)
11. (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.)

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”)

___ Detailed descriptions of system services and constraints that may serve as the basis for a contract.

___ Example would be: "The system shall implement patient privacy provisions as set out in HStan-03-2006-priv."

___ Statements in natural language plus diagrams of system services and constraints, written primarily for customers.

___ Example would be: “Time to restart after failure.”

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

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

___ Example would be: "probability of unavailability is no greater than .005."

___ Example would be: "The automated cooling system must be easy to bypass in case of serious emergencies."

___ A description of how a function could be implemented; intended solely to specify what the function is.

___ Example would be: "In case of a system malfunction, the rocket shall begin a descent to the Earth’s surface at a rate of approximately 9.81m/s^2."

___ Example would be: "The system is to be developed in C++."

___ Example would be: 

**pre-condition:**  
N ≥ 1

**post-condition:** there exists an i in [1,N] such that  
BIG=A[i] & for every j in [1,N], BIG ≥ A[j] & A is unchanged
12. (12 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. Pipe and filter  
- D. Transaction Processing  
- E. Layered architecture  
- F. Client-Server  
- G. Event Processing  
- H. Information systems  
- I. P2P  
- J. Language Processing

___ Commonly used in data processing applications where inputs are processed in separate stages to generate related outputs.

___ Application architecture for the most common type of interactive business systems; user requests for information from (or updates to) a database are processed. Systems are organized in such a way that user actions can’t interfere with each other.

___ Used when the future requirements for user interaction and presentation of data are unknown.

___ Also know as the **abstract machine model**.

___ Used when data in a shared database has to be accessed from a range of locations across a network. May also be used when the load on a system is variable.

___ 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. Systems are often web-based and are accessed through a web browser.

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

___ Also known as a **data-flow architecture**.

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

___ Examples include compilers and command interpreters.

___ Example would be an IDE where each software tool generates design information which is stored in one place and is available for use by other tools.

___ Each level provides a set of services used to implement the next level.
13. System architectures are often modelled using informal block diagrams, as in this figure:

![System Architecture Diagram]

a. (4 pts.) Sommerville notes that Bass, et al., dislike informal block diagrams for describing an architecture. What are the TWO reasons they give in support of their claim that these are poor architectural representations?

b. (4 pts.) What is Sommerville’s view regarding the appropriateness of describing a system architecture using informal block diagrams during the design process? What rationale does he give for his position on this issue? (Be specific.)

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 very general definition of information hiding (i.e., one that is NOT specific to OO design or programming) attributed to David Parnas. What is Parnas’ definition of information hiding?
15. (3 pts.) Two important questions related to whether or not an open source approach should be used when developing software were discussed in class. One concerned the likelihood of volunteer developer involvement. What was the other?

16. (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>Description</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary goal is to convince the supplier of a system (e.g., a product manager) that it is good enough for use (e.g., ready to be delivered to the customer or prepared for sale).</td>
<td>A. Validation testing</td>
</tr>
<tr>
<td>Involves implementing new program functionality <strong>AFTER</strong> implementing and running a test for that functionality.</td>
<td>B. Defect testing</td>
</tr>
<tr>
<td>Also known as <strong>functional testing</strong>.</td>
<td>C. Release testing</td>
</tr>
<tr>
<td>End-user testing performed in the customer environment prior to general release.</td>
<td>D. Alpha testing</td>
</tr>
<tr>
<td>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.</td>
<td>E. Performance testing</td>
</tr>
<tr>
<td>Testing which involves making demands on a system that are outside its design limits.</td>
<td>F. Stress testing</td>
</tr>
<tr>
<td>Concerned with rooting out undesirable system behavior such as system crashes, unwanted interactions with other systems, incorrect computations, and data corruption.</td>
<td>G. Partition testing</td>
</tr>
<tr>
<td>Undertaken to demonstrate that a system meets its requirements.</td>
<td>H. Regression testing</td>
</tr>
<tr>
<td>V&amp;V activity that can consider quality attributes such as compliance with standards, portability, and maintainability.</td>
<td>I. Black-box testing</td>
</tr>
<tr>
<td>May be automated by combining a keystroke recorder and playback tool with a data/output comparator.</td>
<td>J. White-box testing</td>
</tr>
<tr>
<td>Intended to reflect the expected usage of a system in some environment.</td>
<td>K. Operational profile</td>
</tr>
<tr>
<td>Involve people examining a system artifact (requirements, design documents, source code, etc.), usually with the aim of discovering anomalies and defects.</td>
<td>L. Software inspections</td>
</tr>
<tr>
<td>Re-running of one or more tests after some program change that ran without revealing faults prior to the change.</td>
<td>M. Beta testing</td>
</tr>
<tr>
<td>End-user testing performed at the developer’s site.</td>
<td>N. Security testing</td>
</tr>
<tr>
<td>Re-running of one or more tests after some program change that ran without revealing faults prior to the change.</td>
<td>O. Test-driven development</td>
</tr>
<tr>
<td>Re-running of one or more tests after some program change that ran without revealing faults prior to the change.</td>
<td>P. Exhaustive testing</td>
</tr>
</tbody>
</table>
17. (4 pts.) Briefly describe the difference(s) between what Sommerville calls the evolution and servicing life cycle phases of a system.

18. (3 pts.) State Lehman’s “Law of Continuing Change.”

19. (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. Contractual requirements related to the use of modern software engineering techniques, system structure, system documentation, configuration management, and other stabilizing factors can significantly impact maintenance costs.

b. If system maintenance is the responsibility of a company other than the original developer, there may be little or no incentive for a development team to write the software so that it is easy to change.

c. In organizations that still consider development and maintenance to be separate activities, it is normal for development teams to be broken up and for people to work on new projects after a system has been delivered. The new individuals responsible for maintenance must then spend time trying to understand the system before being able to implement changes to it.

d. Unless prior training of maintenance personnel is contractually mandated, a lot of the effort during the maintenance process is taken up with stabilizing the system before changes can be implemented.

e. Maintenance costs can be dramatically reduced if development contracts stipulate that system developers receive group psychotherapy at the very first sign of team instability.

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