You have 90 minutes to work on this exam. It is a "closed-book/closed-notes" test.

**IMPORTANT:** Students who have already taken CEN 4072/6070, Software Testing and Verification, should work PROBLEMS 1-13 AND 18-20 ONLY. All other students should work PROBLEMS 1-17 ONLY. NO CREDIT WILL BE GIVEN for working additional problems.

PRINT your name above NOW and sign the pledge at the bottom of the last page, if appropriate, when you are finished.

PLEASE PRINT ANSWERS IN THE SPACE PROVIDED ONLY – PREFERABLY USING A BALLPOINT PEN TO INCREASE LEGIBILITY. Good luck!

1. (8 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. predicate  
   B. universal quantifiers  
   C. constructor operations  
   D. operational specification  
   E. axioms  
   F. pre-condition  
   G. model-based specification  
   H. post-condition  
   I. assignment function  
   J. component and application assembly  
   K. Lotos  
   L. inspection operations  
   M. function-based specification  
   N. existential quantifiers  
   O. operation signatures  
   P. algebraic specification  
   R. Z  
   S. schema signature

   ___ A specification approach that defines required program behavior in terms of *intended program functions.*
   
   ___ A specification approach whereby a system is specified in terms of a *state model* and operations are defined in terms of *changes to system state.*
   
   ___ Expresses obligatory conditions / relationships among program variables *after* execution in model-based specification.
   
   ___ A specification approach whereby a system is specified in terms of its operations and their relationships via *axioms.*
   
   ___ Defines the interface *syntax* of an object class or abstract data type.
   
   ___ Defines the entities that make up the state of the system being specified using Z.
   
   ___ This specification approach can become cumbersome when object operations are not independent of object state.
   
   ___ Expresses constraints on program variables that an implementer may assume will hold *before* program execution.
2. (3 pts.) Two detailed examples illustrating algebraic specification were given in Chapter 27, "Software Specification." One was “LIST,” which specified a simple FIFO linear list. Identify the other example.

3. (8 pts.) Using pre- and post-conditions, formally specify a program that would satisfy the following requirements:

"The program shall compute the product of two positive integer (input) variables, X and Y, without the use of a multiplication operation. (The program should make use of an addition operation together with a loop, instead.) The result should be stored in (output) variable Z, and the values of X and Y should remain unchanged."

pre-condition:

post-condition:

4. (8 pts.) Provide a function-based specification that is equivalent to the operational program specification given below. (Note that “<>“ means ”NOT EQUAL TO.”)

if X<>0 then
  Y := X+Y
  X := X-(2*X)
end_if

5. (4 pts.) Sommerville identifies a number of “design issues that have to be considered in distributed systems engineering.” Briefly describe the scalability issue, and explain the distinction between “scaling-up” and “scaling-out” in addressing this issue.
6. (10 pts.) Match each application or description below to the **SINGLE MOST APPROPRIATE** type of architecture among the following. (Note: architecture types may apply to none, one, or more than one application or description.)

A. two-tier client/server (C/S) architecture with thin clients
B. two-tier client/server (C/S) architecture with fat clients
C. multi-tier client/server (C/S) architecture
D. distributed component architecture
E. peer-to-peer architecture (P2P)
F. software as a service (SaaS)
G. service-oriented architecture (SOA)
H. master-slave architecture

___ Some argue that this is less intuitive/natural than a client/server architecture, making systems more difficult to visualize, understand, and design.

___ A strategy for designing and building software products through the composition of existing capabilities and services.

___ Suitable for mobile applications where Internet connectivity cannot be guaranteed.

___ Appropriate where the system primarily involves the exchange of information between individual computers on a network and there is no need for this information to be centrally stored or managed (e.g., file-sharing systems).

___ Commonly used to “modernize” legacy systems when separating application processing and data management is impractical.

___ All components are objects that provide services to, and receive services from, other objects; communication is via middleware.

___ Software delivery method whereby a software system is hosted remotely on a provider’s server (a “cloud”).

___ Suitable for applications where both the data and the application are volatile, and for applications where data from multiple sources are integrated.

___ Commonly used in real-time systems where there may be separate processors associated with data acquisition from the system’s environment, data processing, and computation and actuator management.

___ Appropriate for computationally intensive applications for which it is possible to separate the processing required into a large number of independent computations.
7. Sommerville notes that one of the factors that may need to be considered when implementing SaaS is **multi-tenancy**.

a. (4 pts.) From the users’ perspective, what are the two basic goals or objectives associated with *multi-tenancy*?

b. (2 pts.) From a designer’s perspective, what is required in order to achieve these goals/objectives?

8. Consider the diagram below that was used in class to illustrate some of the concepts and terminology used in Aspect-Oriented Software Engineering.

![Diagram of Internet Banking System](image)

a. (3 pts.) Identify all the “core” and “cross-cutting concerns” depicted in the diagram. Clearly annotate the diagram using the labels “core” (for core concerns) and “cross” (for cross-cutting concerns).

b. (3 pts.) Clearly identify, using words, a single, specific example of “tangling” that is represented in the diagram.

c. (3 pts.) Clearly identify, using words, a single, specific example of “scattering” that is represented in the diagram.
9. In his Chapter on Project Management, Sommerville describes four “risk management process” activities: identification, analysis, planning, and monitoring.

   a. (4 pts.) Identify and briefly describe the specific task(s) he associates with risk analysis?

   b. (5 pts.) Identify and briefly describe the specific task(s) he associates with risk planning?

10. (3 pts.) As discussed in class, two common measures of software product reliability are “mean time to system failure” and “probability of no system failure in a specified time interval”. Based on Sommerville’s definition of process reliability, which one of the following would be the most appropriate measure of this process attribute? (Circle ONE only.)

   a. Mean development time to the discovery of a process error
   b. Probability of no process error in a specified development time interval
   c. Mean time required for the process to evolve in order to reflect changing organizational requirements or identified process improvements
   d. Probability that a process error can be avoided or trapped before it results in a product error
   e. Probability that the process can continue in spite of a process error
11. (4 pts.) Which one of the following best reflects the point that Fred Brooks makes with the title of his popular book about being a software project manager? (Circle ONE only.)

a. The title he chose, “The Mythical Mammoth,” refers to an overbearing and threatening project manager, which (he argues) is largely extinct today since project managers now understand that most of their tasks are people-oriented and that poor people skills are an important contributor to project failures.

b. He chose “The Mythical Man-Month” as his title to make the point that not all software engineers are men.

c. The title he chose, “The Mythical Man-Month,” represents the idea that a “man month” – i.e., a person working productively on the same software development task for a month with little or no communication with other team members – is completely unrealistic and therefore “mythical”.

d. He chose “The Mythical Man-Moth” as his title to make the point that the “new-technology man-moth” (which refers to a “half-man, half-moth super hero who can successfully integrate new technology in an organization and then just ‘fly away’ without risk of recidivism”) is just a myth.

e. He chose the title “The Mythical Man-Mouth” to make the point that project managers who communicate with their development staff in a domineering or abusive fashion (Brooks calls such managers “Man-Mouths”) are largely a myth.

f. The title he chose, “The Mythical Man-Month,” refers to a popular measure of effort for software development, the “man-month,” which implies calendar time and people are perfectly interchangeable. The measure, therefore, is “mythical.”
12. (6 pts.) In his Chapter on Process Improvement, Sommerville notes that “Humphrey (1988), in his seminal book on process management, states:

   W.E. Deming, in his work with the Japanese industry after World War II, applied the concepts of statistical process control to industry. While there are important differences, these concepts are just as applicable to software as they are to automobiles, cameras, wristwatches and steel.”

Which ONE of the following best describes Sommerville’s position in this regard?

a. Sommerville disagrees with Humphrey because software quality is not influenced by its manufacturing process but by its design process, where people’s skill and experience are significant.

b. Sommerville disagrees with Humphrey because irrespective of process, if a software project has an inadequate budget or is planned with an unrealistic delivery schedule, product quality will be affected.

c. Sommerville only partially agrees with Humphrey because it is not usually possible to make process improvements that can optimize all process attributes simultaneously in software engineering.

d. Sommerville agrees with Humphrey because measuring the number of product defects allows process modifications that decrease the chances of new defects occurring and increase the chances of detecting those that do occur.

e. Sommerville fully agrees with Humphrey that results from manufacturing engineering can be transferred to software engineering because the process/product relationship is obvious in both contexts.

13. (8 pts.) Consider the following statements related to the CMMI process improvement framework. Circle either "true" or "false" as (most) appropriate. To compensate for random guessing, you will receive +2 pts. for each correct answer and -2 pts. for each incorrect answer. (The minimum score possible for this problem is 0 pts.) Therefore, DON’T CIRCLE AN ANSWER UNLESS YOU ARE MORE THAN 50% SURE THAT IT IS CORRECT!

   a. Predecessors of the CMMI process improvement framework include the SEI’s Software CMM, the People CMM, and the Systems Engineering CMM. true false
   
   b. A CMMI assessment involves examining software processes in an organization and rating them on a six-point scale that assigns a level of maturity to each process area. true false
   
   c. The staged CMMI model is used to assess the processes employed in specific software development stages, while the continuous model is used to measure the maturity of an organization’s software processes as a continuous whole. true false
   
   d. CMMI generic goals and practices are associated with the institutionalization of good practice rather than being technical in nature. true false
14. (10 pts.) Consider each of the following assertions and circle either “true” or “false” as appropriate. (Note: “|x|” refers to the absolute value of variable x, and “<>” means “NOT EQUAL TO.”) To compensate for random guessing, you will receive +2 pts. for each correct answer and -2 pts. for each incorrect answer. (The minimum score possible for this problem is 0 pts.) Therefore, DON’T CIRCLE AN ANSWER UNLESS YOU ARE MORE THAN 50% SURE THAT IT IS CORRECT!

a. \{y=1\} \text{input}(x); \ y := x-1 \ \{y\leq x\} \quad \text{true} \quad \text{false}

b. \{x<0 \land x=-5\} \text{if} \ x<5 \ \text{then} \ x := x-1 \ \{x\geq 0 \lor |x|\leq 6\} \quad \text{true} \quad \text{false}

c. \{x=y^k\} \ y := y+1 \ \{x=(y+1)^k\} \quad \text{true} \quad \text{false}

d. \{x=17\} \text{while} \ x<>5 \ \text{do} \ x := x+1 \ \{x=5\} \quad \text{true} \quad \text{false}

e. \{x>0\} \ s1 \ \{y=x\} \Rightarrow \{x>0 \land y=5\} \ s1 \ \{y\leq x\}\quad \text{true} \quad \text{false}

15. (10 pts.) Use the WEAKEST PRECONDITION-BASED METHOD to prove the following assertion. SHOW ALL STEPS as illustrated in class.

\{y=17\} \text{if} \ y>0 \ \text{then} \ y := y-5 \ \{y\geq 0\}
NOTE: if you have already taken CEN 4072/6070, Software Testing and Verification, DO NOT work the problems on this page! (Work problems 18-20 instead.)

16. a. (3 pts.) Give the antecedents ("initialization, preservation, and finalization") for the while loop Rule of Inference (ROI): (Fill-in the blanks.)

   ______________ , ______________ , ______________
   ---------------------------------------------------------------
   \{P\} while b do S \{Q\}

b. (8 pts.) Prove the following assertion of weak correctness using the while-loop ROI with the given invariant, I. Assume that all program variables represent INTEGER values. Show ALL steps as illustrated in class.

   \{ true \} \quad \text{Use I: } x \geq 5

   x := 11
   \text{while } x > 5 \text{ do}
   \quad x := x - 1
   \text{end\_while}

   \{ x = 5 \}

------------------------------------------------------------------- please do not write below this line ---------------------------------------------------------------------
NOTE: if you have already taken CEN 4072/6070, Software Testing and Verification, DO NOT work the problems on this page! (Work problems 18-20 instead.)

17. (8 pts.) Prove $f = [P]$ where $f = (x > 0 \rightarrow x, y := x-3, x+3 \mid x \leq 0 \rightarrow x, y := x, y)$ and $P$ is the program:

if $x > 0$ then
    $y := x+3$
    $x := x-3$
end_if

Show all steps as illustrated in class.
NOTE: problems 18-20 are for students who have already taken CEN 4072/6070, Software Testing and Verification, ONLY! All other students should work problems 14-17 instead.

18. (12 pts.) Match each description below to the **SINGLE MOST APPROPRIATE** approach for supporting software reuse among the following. (Note: reuse approaches may apply to none, one, or more than one description.)

A. Application frameworks
B. WAFs
C. Software product lines
D. COTS-solution systems
E. COTS-integrated systems
F. ERP systems
G. Configurable vertical applications
H. Legacy system wrapping
I. Program libraries
J. Model-driven engineering
K. Design patterns
L. Architectural patterns
M. Program generators
N. AOSD

____ Single product systems sold in the commercial marketplace based around a generic solution and standardized processes. The *vendor* is responsible for maintenance and provides the platform.

____ Collections of concrete and abstract objects that are designed for reuse through specialization and the addition of new objects. They usually incorporate good design practice through design patterns.

____ Large scale integrated systems designed to support business practices such as ordering and invoicing, inventory management, and manufacturing scheduling. They are created by configuring a generic system with information about a company’s business processes and rules.

____ Related applications that are developed from one or more base applications. A generic system is adapted and specialized to meet specific requirements for functionality, target platform, or operational configuration.

____ Two or more product systems sold in the commercial marketplace (perhaps by different vendors) are used together to provide customized functionality. The *owner* is responsible for maintenance and provides the platform.

____ Can support the construction of dynamic websites. Its architecture is usually based on the Model-View-Controller composite pattern, and it usually incorporate one or more specialized frameworks that support specific application features.

19. (7 pts.) Briefly explain what is meant by “Inversion of Control” (IOC) in the context of application frameworks, and give a concrete example that clearly illustrates the idea.
NOTE: problems 18-20 are for students who have already taken CEN 4072/6070, Software Testing and Verification, ONLY! All other students should work problems 14-17 instead.

20. (20 pts.) Match each description below to the **SINGLE MOST APPROPRIATE** configuration management term among the following. (Note: terms may apply to none, one, or more than one description.)

- A. SCI  
- B. Baseline  
- C. Workspace  
- D. CAB  
- E. Merging  
- F. Version  
- G. Trunk  
- H. Codeline  
- I. Lock  
- J. Capture  
- K. Variation  
- L. Archive  
- M. Temporary variant  
- N. Mainline  
- O. Allocation  
- P. Release  
- Q. System building

___ Anything associated with a software project (design, code, test data, document, etc.) that has been placed under configuration management.

___ An instance of a configuration item that differs, in some way, from other instances of that item.

___ A collection of component versions that make up a system. The versions cannot be changed, so it should always be possible to re-create the collection.

___ A set of versions of a software component and other configuration items on which that component depends.

___ A sequence of baselines representing different versions of a system.

___ A version of a system that has been delivered to customers (or other users in an organization) for use.

___ A private development area where software can be modified without affecting other developers who may be using or modifying that software.

___ The creation of a new codeline from a version in an existing codeline. The new codeline and the existing codeline may then develop independently.

___ The creation of a new version of a software component by combining separate versions in different codelines.

___ The creation of an executable system version by compiling and linking the appropriate versions of the components and libraries making up the system.

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