1. a. (6 pts.) Briefly compare and contrast the use of redundancy and diversity in software systems that are designed for dependability. Explain how these different techniques are intended to work.

b. (4 pts.) What specific strategies are normally employed in computer-based systems (i) where availability is a critical requirement (e.g., in e-commerce systems), and (ii) to provide resilience against external attacks intended to exploit a common vulnerability?

2. a. (3 pts.) What is an operational failure in the context of dependable systems engineering? Be specific.

b. (4 pts.) What, according to Sommerville, are the two important design goals of a dependable operational process?
3. (4 pts.) Sommerville uses the European Space Agency’s experience with the Ariane 5 rocket to illustrate a specific shortcoming in dependable systems engineering. Briefly describe the experience and explain what the specific shortcoming was.

4. (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.)

<table>
<thead>
<tr>
<th>A. Availability</th>
<th>E. Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Reliability</td>
<td>F. Repairability</td>
</tr>
<tr>
<td>C. Safety</td>
<td>G. Maintainability</td>
</tr>
<tr>
<td>D. Security</td>
<td>H. Error tolerance</td>
</tr>
</tbody>
</table>

___ It must be possible to diagnose a problem, access the component that has failed, and make changes to fix that component quickly.

___ A judgment of how likely it is that the system will not cause damage to people or its environment

___ Software can be adapted economically to cope with new requirements; there is a low probability that making changes will introduce new errors into the system.

___ The probability that the system will correctly deliver services as expected by users

___ A judgment of how likely it is that the system can resist accidental or deliberate intrusions

___ Can be considered as part of usability; when user errors occur, the system should, as far as possible, detect these errors and either fix them automatically or request the user to re-input the data.

___ The probability that a system will be up and running and able to deliver useful services to users

___ A judgment of how well a system can maintain the continuity of its critical services in the presence of disruptive events such as equipment failure and cyberattacks
5. Sommerville points out the designs used in the Airbus 340 and Boeing 777 flight control systems as examples of two very different approaches to successfully achieving dependability.

a. (4 pts.) Describe the specific approaches used in each design.

b. (3 pts.) Explain the dependability-related rationale for the approach used by Boeing.

6. Provide formal specifications via pre- and post-conditions for the program behaviors described below. Assume that N is an integer constant.

a. (3 pts.) Perform integer subtraction using the arithmetic primitive "subtract 1" and a repeat until loop. Let M be the minuend, S be the subtrahend, and D be the difference. Assume that the subtrahend is positive.

pre-condition:
post-condition:

b. (9 pts.) Set variable I to the index of the last instance of Y in the non-empty array A[1:N], or to -1 if Y is not in the array.

pre-condition:
post-condition:

c. (8 pts.) Set boolean variable PERM to True if non-empty array A[1:N] is a permutation of array B[1:N], and to False otherwise. A permutation of a list of elements is a list comprised of the same (and ONLY the same) elements in a (possibly) different order. (For example, sorting a list of elements always yields a permutation of the original list.) Make use of (but do NOT formally specify) the function Count in your specification, where Count(e,K) is the number of occurrences of element e in array K[1:N].

pre-condition:
post-condition:
7. (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.) Place the letter of ONE TERM to the **left** of each description.

- A. predicate
- B. universal quantifiers
- C. constructor operations
- D. operational specification
- E. axioms
- F. pre-condition
- G. model-based specification
- H. post-condition
- J. assignment function
- K. Lotos, RSL/RSA
- L. inspection operations
- M. function-based specification
- N. existential quantifiers
- O. operation signatures
- P. algebraic specification
- R. VDM, B
- S. schema signature

___ A specification approach whereby a system is specified in terms of its operations and their relationships via *axioms*

___ Defines the entities that make up the state of the system being specified using Z

___ Defines the interface *syntax* of an object class or abstract data type

___ A specification approach that defines required program behavior in terms of *intended program functions*

___ Relate the operations used to construct entities of the defined sort with operations used to inspect its values

___ Used to specify program data mappings; its domain corresponds to the initial data states that would be transformed into final data states by a suitable program.

___ Mature notations for developing model-based specification

___ That part of a schema which defines conditions that are always true

___ Used in model-based specification to express obligatory conditions / relationships among program variables *after* execution

___ Used to assert that some predicate holds FOR AT LEAST ONE or FOR SOME member of a given set

___ This specification approach can become cumbersome when object operations are not independent of object state.

___ 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*

___ Guttag first discussed this approach for the specification of abstract data types. Cohen, et al., showed how the technique can be extended to complete system specification using an example of a document retrieval system.

___ Expresses constraints on program variables that an implementer may assume will hold *before* program execution
The following is an excerpt from Sommerville’s Chapter 27 example illustrating how formal specification can be used to specify an interface in a critical system specification.

\[
\text{Enter}(S, CS, H) = \\
\quad \text{if } \text{In-space}(S, CS) \text{then } S \text{ exception} \\
\quad \text{elsif } \text{Occupied}(S, H) \text{then } S \text{ exception} \\
\quad \text{else } \text{Put}(S, CS, H)
\]

a. (2 pts.) What formal specification technique does Sommerville use in this example?

b. (2 pts.) Briefly describe the critical system being specified.

c. (3 pts.) What specific system function does the Enter operation above specify?

9. (12 pts.) Consider each of the following assertions and circle either “true” or “false” as appropriate. (Note: \(|x|\) refers to the absolute value of variable \(x\).) 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 problem.

a. \{true\} while \(|x| = 5\) do \(x := x - 2x\) \(\{x=17 \lor \text{true}\}\) \hspace{1cm} true \hspace{1cm} false

b. \[\{x>0\} S \{x=17\}\] \(\Rightarrow \[\{x\geq0\} S \{x\geq0\}\]\) \hspace{1cm} true \hspace{1cm} false

c. \(\wp(A := \text{true}, A \land B \land C \land D) = (\neg A) \land B \land C \land D\) \hspace{1cm} true \hspace{1cm} false

d. \(\{a<xy+2x\} a := a-x; y := y+1 \{a\leq xy\}\) \hspace{1cm} true \hspace{1cm} false

e. \(\wp(\text{paint the living room ceiling, the house is all painted}) = \text{The whole house, except for the living room ceiling, is all painted.}\) \hspace{1cm} true \hspace{1cm} false

f. \(\wp(\text{FOUND} := \text{false}, [\text{FOUND} \land \exists 1\leq i \leq N \mid J=i \land \text{LIST}[J]=\text{KEY}) \lor \neg \text{FOUND}]) = \forall 1\leq k \leq N, \text{LIST}[k] \neq \text{KEY}\) \hspace{1cm} true \hspace{1cm} false
10. a. (3 pts.) Give the antecedents ("initialization, preservation, and finalization") to complete the while loop Rule of Inference (ROI):

\[
\{P\} \text{ while } b \text{ do } S \{Q\}
\]

b. (8 pts.) Prove the assertion of weak correctness below using the While-Loop Rule of Inference with the invariant: \( p=2^{-k} \land k \leq n \). SHOW AND JUSTIFY ALL STEPS AND CASES AS ILLUSTRATED IN CLASS.

\[
\{\text{n}\geq\text{0}\}
\begin{align*}
p &:= 1 \\
    k &:= 0 \\
    \text{while } k < \text{n do} \\
    & \quad p := p/2 \\
    & \quad k := k+1 \\
    \text{end\_while}
\end{align*}
\]

\[
\{p=2^{-n}\}
\]
11. Recall the ROI for the if-then statement derived in class:

\[
\{P \land b\} \ S \ \{Q\}, \ (P \land \neg b) \implies Q
\]

\{P\} \text{ if } b \text{ then } S \ \{Q\}

a. (2 pts.) Extrapolating from that derivation, give the antecedents to complete the Rule of Inference for the if-then-else statement:

\[
\{P\} \text{ if } b \text{ then } S_1 \text{ else } S_2 \ \{Q\}
\]

b. (8 pts.) Use this ROI to prove the assertion:

\[
\{\text{TEMP=17}\} \text{ if } A > B \text{ then } Z := A \text{ else } Z := B \ \{Z=\text{MAX}(A,B) \land \text{TEMP}=17\}
\]

SHOW AND JUSTIFY ALL STEPS AND CASES.
12. (12 pts.) Prove $f = [P]$ where $f = (x \geq y \rightarrow x, y := y, x \mid x < y \rightarrow I)$

    and $P$ is:  if $x > y$ then
     $x := x + y;$
     $y := x - y;$
     $x := x - y$
    end_if

STATE AND PROVE ALL CONDITIONS, STEPS, AND CASES.

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