CS445 / CS645 / ECE451
Fall 2016 — Final Exam

21 December 2016, 4:00pm–6:30pm

Instructor: Daniel M. Berry

Time allowed: 2.5 hours = 150 minutes

No aids allowed (*i.e.,* closed book).

Answer all of the questions on this exam paper.

There are 9 questions for a total of 150 marks.

Plan your time wisely: 1 minute per mark

____

Your Name and Student Number

In the immortal words of the yet to be born Jean-Luc Picard of Earth,

Make it so!

Q1		scaled to 18
Q2		scaled to 15
Q3		scaled to 15
Q4		scaled to 15
Q5		scaled to 09
Q6		scaled to 18
Q7		scaled to 05
Q8		scaled to 25
Q9		scaled to 30
TOTAL		scaled to 150

In this exam, a short underscore of 1 inch (= 2.54 cm) should be filled with one word. A long underscore of 3 inches (= 7.63 cm) should be filled with a phrase consisting of one to several words. In the either case, if you cannot think of exactly the right number of words, then give the best answer that you can and we'll give it as many marks as we can, possibly even full credit! If you cannot even think of just words to fill in, then write an answer as a sentence, and we'll give it as many marks as we can.

"(complete the sentence)" means that in the space following the current line, you are to complete the sentence that was started and interrupted with the "(complete the sentence)"

If you are asked a question, which ends with a "?", you are to answer that question in the space following the current line.

In this exam, if you are asked for a simple answer, you need not justify it, unless you are also asked explicitly "Why?". However, you may always write down assumptions that can help us give you partial credit.

In the exam questions,
"CBS" means "computer-based system".
"NFR" means "non-functional requirement", a.k.a. "quality attribute".
"RE" means "requirements engineering".
"SRS" means "software requirements specification, written according to some standard, e.g., IEEE".
"UM" means "user's manual".

1. **[18 total marks]** RE Reference Model and Validation

One example we beat to death in class when talking about the RE reference model was a traffic light. In this example, we observed a case in which a correct specification of a desired CBS is not strong enough to entail the CBS's requirements. Specifically a correct specification of a traffic light is not strong enough to entail the holding of the requirement of no perpendicular collisions:

$\mathsf{TrafficLightSpecification} \nvdash \mathsf{NoPerpendicularCollisions}$

Instead, domain assumptions about drivers obeying the law and cars obeying drivers were needed:

$(\mathsf{DriversObeyLaw} \wedge \mathsf{CarsObeyDrivers}), \mathsf{TrafficLightSpecification} \vdash \mathsf{NoPerpendicularCollisions}$

In class, we considered the possibility of strengthening the specification of a traffic light so that DriversObeyLaw is not needed to be able to entail the requirement, *i.e.,*

$(\mathsf{CarsObeyDrivers}), \mathsf{TrafficLightSpecification}' \vdash \mathsf{NoPerpendicularCollisions}$

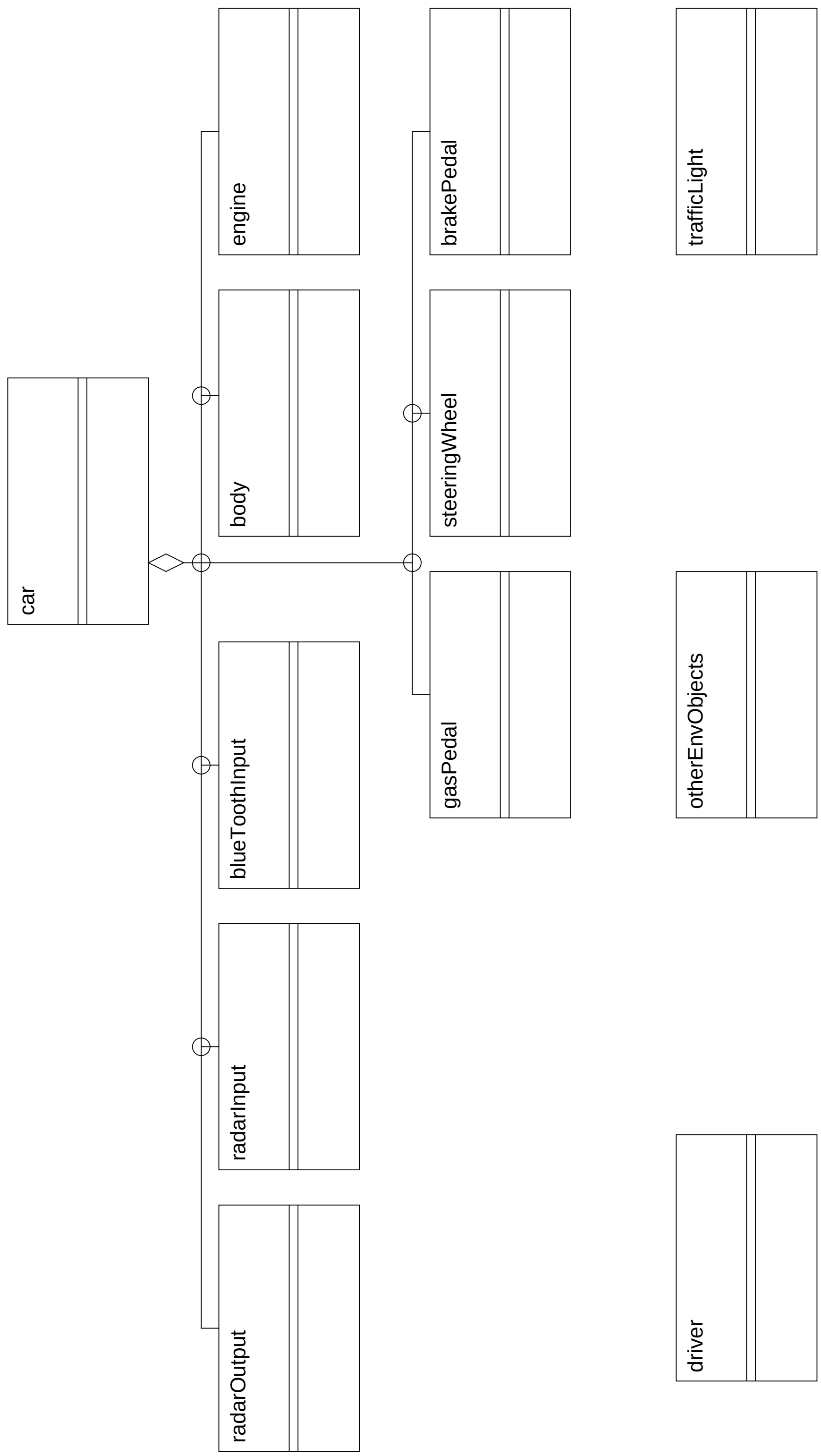
One suggested strengthening was to have traffic lights send a Bluetooth message saying "Stop!" to every car near enough to the intersection that would be facing a red light. This idea would work very nicely with the self-driving cars that we hear about a lot these days!

Below is the skeleton of a UML class model of a self-driving car's world. A car has a body and an engine, and it has a gas pedal, a steering wheel, and a brake pedal that are usable by both the car's software and a human driver that is sitting in the car's driver's seat. A car has also a radar output that sends out sonar that gets reflected back by other environmental objects and picked up by the car's radar input. Finally, a car has a Bluetooth input that is prepared to receive a message from traffic light to stop the car.

The skeleton already shows the aggregation links identifying components of car. The little circle at the intersection of two links says that the links *are* connected. This way, it is easy to interpret a non-circled intersection of links as meaning that the links are *not* connected.

In this skeletal diagram, please

- fill in all and only the other required links,
- mark each class as being in the environment with an "E", being in the system with an "S", or being in both with a "E S",
- put each of the following operations in the right classes (Don't worry if an occasional fill-in oversteps its box boundary, so long as most if it is inside.): hearReflectedSonar, letUp (opposite of push), push, reflectSonar(S), sendOutSonar, stopTheCar, turnLeft, and turnRight (Note that each operation could be in more than one class.), and
- mark each actor of the car system with the stereotype "≪actor≫".


car
engine
body
blueToothInput
radarInput
radarOutput
brakePedal
steeringWheel
gasPedal
trafficLight
otherEnvObjects
driver

Any model of the real world on which D and R in $D, S \vdash R$ are based will be inaccurate for at least one of the following reasons:

- Each modeling notation in which the model is expressed has different _______________ and _______________.
- The model must trade off _______________ with _______________.
- The model is based on a scientific theory that is known to be _______________ or to have _______________.
- There are *always* _______________ to the model that we do not know about, because we do not fully _______________ the real world.

Therefore, as a formula, $D, S \vdash R$ is inherently _______________.

2. **[15 total marks]** UML Domain and Use Case Models

   Consider the following famous nonsense sentences that follow English grammar rules

   > The gostak distims the doshes.
   > The doshes are galloons.
   > Therefore, some galloons are distimmed by the gostak.
   > The gostak distunks the glauds.
   > The gostak distunks the whole delcot of doshes, a delcot being where doshes deave.
   > The drokes descren the whole delcot of doshes.
   > No glaud will vorl the doshes from the gostak.

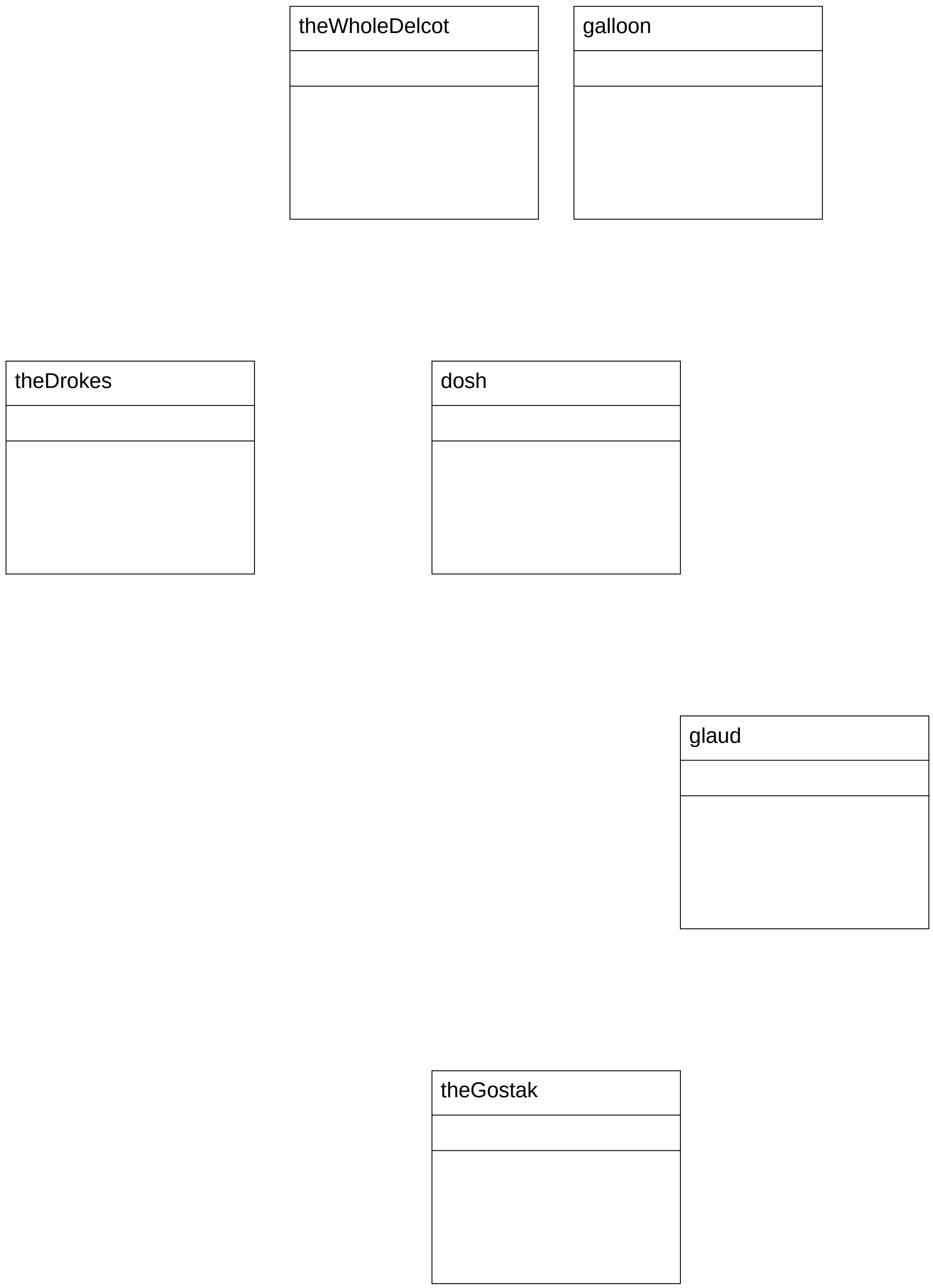
   On the next page you will find the skeleton of a UML class diagram of the world described by the nonsense sentences. The diagram shows only the classes. Missing are stereotypes, multiplicities, links, and methods (*i.e.,* operations). Please complete the diagram with the missing stereotypes, multiplicities, links, and methods. While there is space in the diagram for attributes, you are not to show any.

   You must insert "≪actor≫" in each class that describes an actor.

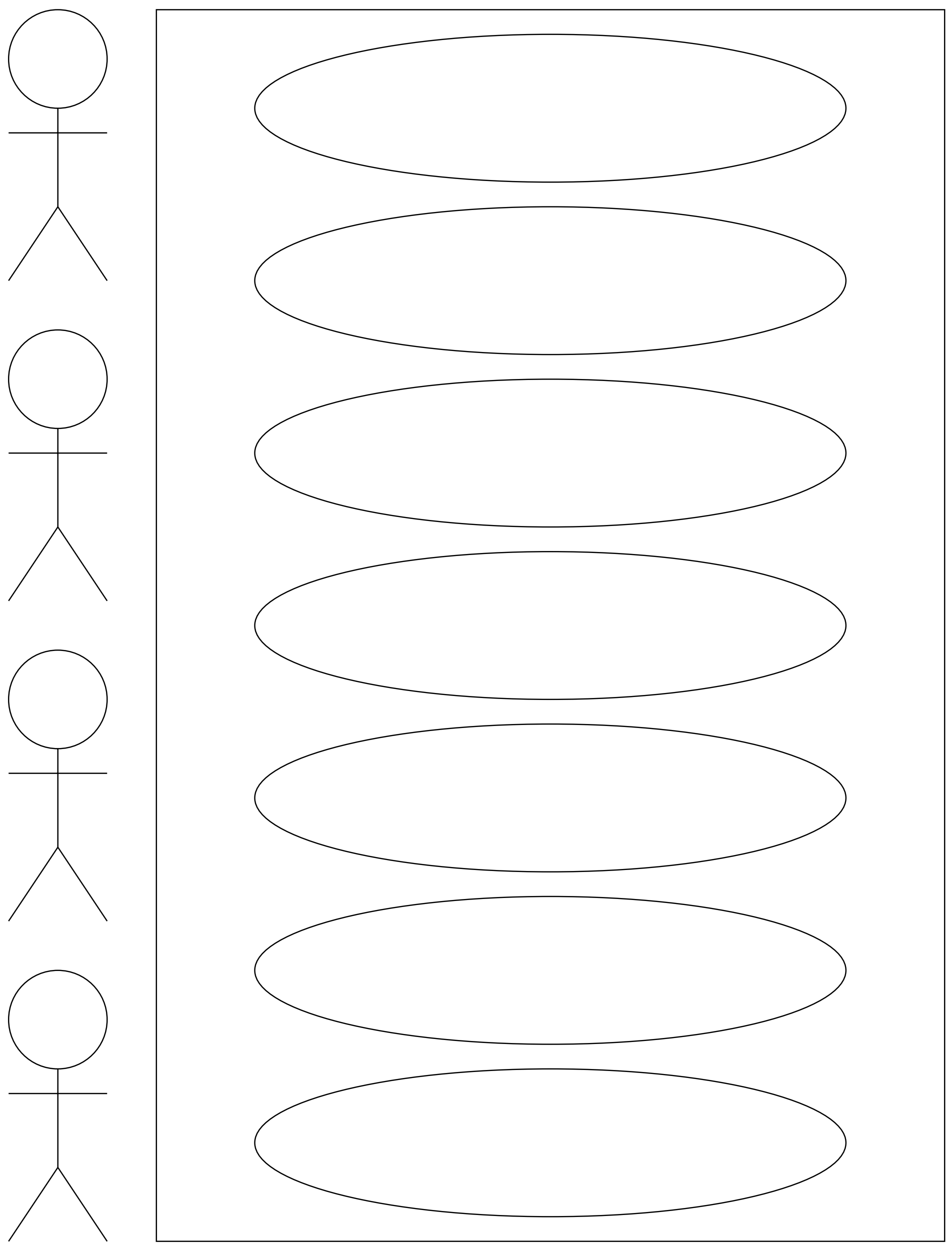
   You must insert all and only links that are needed, including those describing aggregation (componentry) and inheritance (subclassing).

   You must place each of the following verbs in only the right classes: deave, descren, distim, distunk, and vorl. It is possible that some appear in more than one class.

   You must provide the right multiplicity to each class (but not to the links), using "1" if there is exactly one instance of the class and "*" if there are an unknown, possibly multiple number of instances of the class.

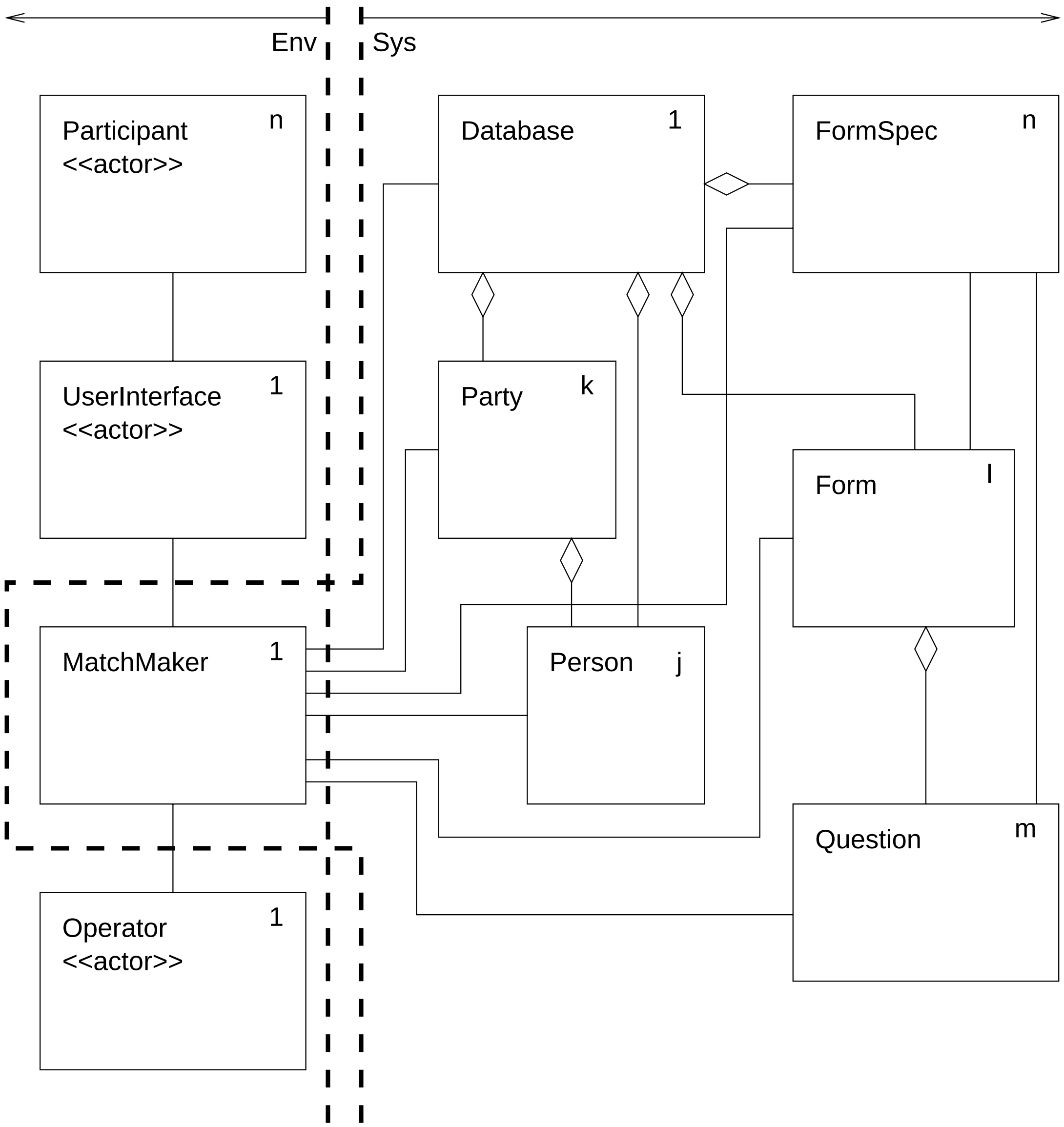

theWholeDelcot
galloon
theDrokes
dosh
glaud
theGostak

Below is a skeleton of a use case diagram for the same world of nonsense sentences. It has more actors and blank use cases than are needed. Label each actor that is needed with the name of an actor from the class diagram, and fill in each blank use case that is needed with the name of an operation from the class diagram that is visible to an actor by virtue of the links in the class diagram. Draw a link from each actor to only each use case that it uses by virtue of the nonsense sentences. It is OK to have use cases that are not linked to any actor. Finally, put an "X" through each actor and blank use case that is not needed.



3. **[15 total marks]** Matchmaker

Consider the main domain model for Matchmaker,



(a) What are the effects on the behavior of class MatchMaker of the decision to have another program, UserInterface, interact with the Participants?

i. MatchMaker does not have to (Your answer must be significantly more detailed than "interact with the Participants".) (complete the sentence)

ii. MatchMaker can assume that (complete the sentence)

(b) Prior to the delivery of Deliverable 5, the amended project vision document included the following description of a second kind of question in a Matchmaker questionnaire:

2. a question to be answered only once; however the question itself says whom it is about: the user or his or her ideal mate:
   - part I of the 1966 version of the Operation Match questionnaire and
   - the entirety of the 1967 version of the Operation Match questionnaire

   Each such question may be linked to another question as its mirror question; that is, one of the mirrors is about the user and the other is about the user's ideal mate....

After the delivery of Deliverable 5, the customers removed the requirement for Matchmaker to deal with this kind of question in the questionnaire.

i. More than envisioned in the amended vision document, dealing with this kind of question required that Matchmaker have (complete the sentence)

ii. To implement this requirement, the amended vision document had suggested to use (complete the sentence)

iii. Which data structure in the main domain model is there to support this suggestion?

iv. Once this requirement is gone, the only kind of questions left are (complete the sentence)

v. Finally, why was this requirement removed from the project's vision document for Deliverable 6?

(c) The definition of good and balanced match, also known as balanced mutual match and mutual balanced match, that you were required to have somewhere in your UM or SRS is allowed to use any data that appears in a ________________ because these data consists of information that is ________________ to the ________________.

(d) More generally speaking, a definition in the UM or SRS may use terms from the vocabulary of the (complete the sentence)

(e) Specifically, according to the vision document and its updates, the definition of good and balanced match, also known as balanced mutual match and mutual balanced match, needs to use what three data items about each question in a participant's questionnaire?

i.

ii.

iii.

4. **[15 total marks]** Graduate Student Lectures

(a) Jia Wu described the development of software for a scientific application ALDEx2, to help with RNA sequencing. The developers did not appear to have a defined, systematic requirements analysis phase, as this course suggests. Rather, requirements seem to have been determined on the fly as they were needed to write the code.

Among the lessons learned that Jia reported were:

i. Research is fluid; therefore requirements ____.

ii. The simplest statement of *abstract requirements* tuns out to be the scientific ______________ statement.

iii. The developers of ALDEx2 ended up ______________ requirements from Geology and Biology.

iv. The main non-functional requirement (NRF) centered around the need for the ALDEx2's customers, as academics, to ______________ their results.

(b) Diya Burman reported on the Y2K Millenium Bug and the Ariane 5 Rocket Explosion.

The Y2K Millenium Bug was simply that in the early days of computing, in the 1950s and 1960s, to conserve the very expensive and very limited (e.g., 128K words) memory space, the year part of a date was routinely expressed as only two digits, with the assumption that the missing, century digits were "19". It was always known that in the year 2000, this assumption would cease to be true. Year-by-year, it was becoming more and more necessary to invest the very non-productive time to change the ever-growing body of existing, legacy software to use four digits for each date. It was true also that because of Moore's law, year-by-year, the need to conserve memory became less and less.

Give two main reasons that an organization does not want to change its existing, legacy software on which the organization depends.

i. The cost to change all this software is ______________.

ii. With each change, a program becomes more and more ______________.

The developers of the Ariane 5 rocket reused a lot of already fully-debugged code from the development of the outwardly similar Ariane 4.

This reuse of code is normally a ____. However, such reuse must be accompanied by ______________ and ______________ that the ______________ of the original code still ______________.

(c) Many an organization installs a variation of SAP or Oracle, each being a large package of already developed and debugged applications for a variety of organizational processes. Each application and a collection thereof can be installed in any of many available configurations that should be set to the client's needs.

Evan Citulski talked about two failed deployments, one of SAP and one of Oracle, in governmental organizations. The main less on learned from these failed deployments is that:

> While the deployed code is fully debugged and meets its its requirements, a configuration file is so complex with so many choices and options that depend on each other, that a configuration file

must be considered to be a ______________ in its own right, for which it is necessary to do

full ______________________________________.

Many a customer who has licensed SAP or Oracle *believes* that since the code is, as emphasized by SAP's or Oracle's sales force, already developed and fully debugged,

full ______________________________________ is not ______________.

(d) Applying the stable marriage algorithm (SMA), presented by Haiyu Zhen, or the stable roommates algorithm (SRA), presented by Kshitiz Thapa, requires being able to determine for each person, A, and potential matches, B and C, for A, which of B and C is ______________ by A.

Using these algorithms during the execution of Matchmaker would make use of the definition

of ______________________________________ that must be in the Matchmaker requirements specification.

In the table below, check a table cell if the algorithm at the top of the cell's column is directly applicable to determining matches for the mode and variation at the left of the cell's row.

Mode and Variation	SMA	SRA
Dating Mode		
Party Mode, Bipartite Party		
Party Mode, Unipartite Party		

5. **[9 total marks]** NFRs

(a) Why are the NFRs robustness and testability in conflict with each other?

(b) The first step in every method to validate, *i.e.,* test, that a not directly measurable NFR has been met by a CBS's implementation is to get the ______________ to ______________ that the method's indirect measure for the NFR is ______________ as a ______________ for satisfaction of the NFR by the CBS.

(c) One method, M1, of ranking n requirements by priority is to assign to each requirement one of four ranks: critical, necessary, optional, and useless. (FYI, these are intended to be in decreasing importance.)

Another method, M2, is to compare every pair of requirements to decide which of the two has the higher priority. If the pair-wise rankings are not inconsistent ("$A > B$", "$B > C$", and "$C > A$" *are* inconsistent.), then it is possible to compute a total ordering of the n requirements.

Fill in the table below to show the trade off between these two methods. In this table, each cell has a numerical value that may be expressed as a constant or as a formula that involves n:

Method	Number of Assignments or Comparisons	Number of Different Priority Ranks
M1		
M2		

So, M1 and M2 really trade ______________ of the ranking with ______________ of the ranking.

6. **[18 total marks]** Ambiguity

(a) Consider the 10 variations of the sentence I want to date Joe., obtained by distributing only and also to different places in the sentence.

_______ 1. Only I want to date Joe.

_______ 2. I only *want* to date Joe.

_______ 3. I want only to date Joe.

_______ 4. I want to only date Joe.

_______ 5. I want to date only Joe.

_______ 6. Also I want to date Joe.

_______ 7. I also *want* to date Joe.

_______ 8. I want also to date Joe.

_______ 9. I want to also date Joe.

_______ 10. I want to date also Joe.

Items B through M below are sentences, each of which might clarify, interpret, explain, or exemplify the meaning of one of 10 sentences above. Item A is simply the claim that a sentence above is classified as meaningless. Please, in the underscore next to each sentence S above, write the letter labeling every of the 13 items below that classifies, interprets, explains, or exemplifies S. Each letter appears in the underscores at most once, but the underscores of one of the sentences above gets two letters, and some letters may not be used at all.

A. Meaningless
B. No one else wants to date Joe.
C. I don't want anything else.
D. I don't want to date also Jim.
E. I don't want to marry Joe.
F. I *have* a date with Joe.
G. Everyone else wants to date Jim.
H. I have been dating Jim.
I. I have been just friends with Joe.
J. I will not *actually* date Joe.
K. Jim and Joe are friends.
L. Let's see what I want. I want to ski.
M. Mary and Thelma want to date Joe.

(b) The sentence

The sensors issue an alarm when they detect an incursion.

has at least four different meanings. By careful use of a singular, possibly collective, subject, rewrite this sentence four different ways to illustrate four of these meanings.

i.

ii.

iii.

iv.

7. **[5 total marks]** Cost Estimation

(a) It is surprising at first glance that the COCOMO formula for the duration, D of a software development project, in units of months

$D = c \times E^d$

as a function of the effort, E of the project, in units of person months lacks a term whose unit

is ________________.

(b) The COCOMO formula for the effort, D of a software development project, in units of person months as a function of the size, S of the project's code, in units of thousands of lines of code (KLOC), is

$E = a \times S^b \times X$

The exponent b of S in all versions of the formula is greater than 1 ($b > 1$). Therefore, effort grows

________________ than linearly with increasing size S of the code, because connections between software

components grow ________________ with increasing size S of the code.

(c) It was observed a long time ago that a programmer produces the same number of debugged lines of code per day in any programming language in which he or she is competent. That is, a programmer, who is competent in both Assembly and Lisp, that writes 20 debugged lines of Assembly code per day, writes 20 debugged lines of Lisp per day. Therefore, the COCOMO mapping from S, program size, to E, effort is

not ________________ on the ____ used in the project.

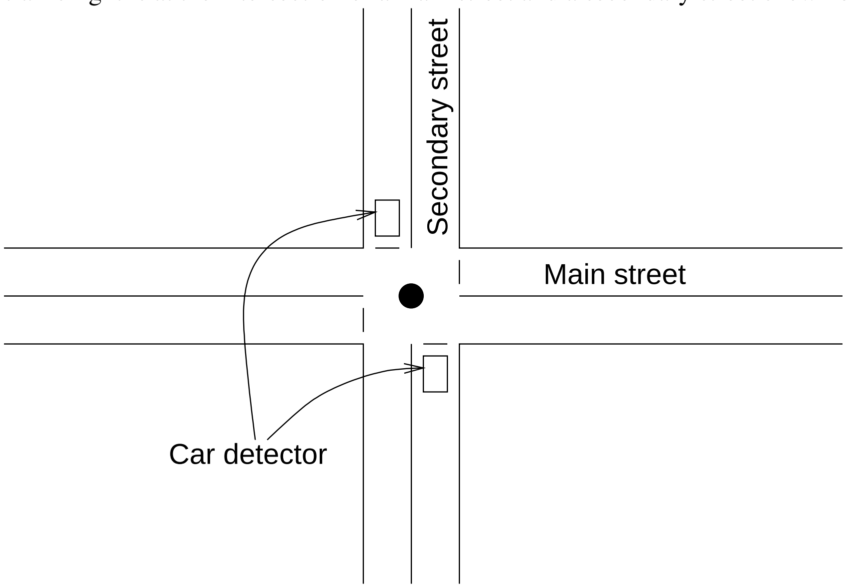
(d) It has been said that in every large system SYS, there is a small program p struggling to get out. That is, 10% of the code for SYS is p, and the remaining 90% of the code is for checking input, dealing with exceptions, dealing with other systems, being robust, etc. This observation calls to mind Fred Brooks's

observation that building a ____ costs ________________

times the cost of building its ____.

8. **[25 total marks]** State Machines and Linear Temporal Logic

You will be playing with a linear temporal logic and an state machine specification of a simple traffic light. The traffic light is at the intersection of a main street and a secondary street shown below:



The secondary street has behind its two stop lines car detectors each of which senses the presence of any car idling above it while stopped for a red light. The traffic light ensures that at any time, either only its two main street faces show green or only its two secondary street faces show green. In each face, the cycle is red, green, yellow, and back to red. As long as there are no cars idling above the two secondary street car detectors, the main street faces continue to show green. As soon as at least one car is idling above the two secondary street car detectors, the traffic light begins the cycles to end up showing green in the secondary street faces and showing red in the main street faces. The traffic light continues to show green in the secondary street faces for a time period known as the long time out (LTO), at which time the light cycles back to continuously showing green on its main street faces until such time as at least one car is idling above the two secondary street car detectors. In the cycling, each transition, e.g., such as from showing yellow to showing red, that is not governed by the long time out or the car detectors, is for a time period known as the short time out (STO). You will be considering a linear temporal logic and a finite state machine specification of this behavior. Each of the specifications has states and events. In these specifications, the two time outs are caused by an external machine not described in the specifications.

In the states,
M means “main street faces show”,
S means “secondary street faces show”,
R means “red”,
G means “green”,
Y means “yellow”,
T means “timed”, and
NT means “not timed”.
Among the events,
D>=1C means “car detectors detect at least one car”,
D0C means “car detectors detect zero cars”,
STO means “short time out”, and
LTO means “long time out”.

Consider the following specification written in linear temporal logic:

$\Box(\mathsf{MGSR\text{-}NT} \Rightarrow (\mathsf{MGSR\text{-}NT}\ \mathcal{W}\ (\mathsf{D{>}{=}1C})))$
$\Box((\mathsf{MGSR\text{-}NT} \wedge \mathsf{D{>}{=}1C}) \Rightarrow \bigcirc\ \mathsf{MYSR})$

$\Box(\mathsf{MYSR} \Rightarrow (\mathsf{MYSR}\ \mathcal{W}\ (\mathsf{STO})))$
$\Box((\mathsf{MYSR} \wedge \mathsf{STO}) \Rightarrow \bigcirc\ \mathsf{MRSR})$

$\Box(\mathsf{MRSR} \Rightarrow (\mathsf{MRSR}\ \mathcal{W}\ (\mathsf{STO})))$
$\Box((\mathsf{MRSR} \wedge \mathsf{STO}) \Rightarrow \bigcirc\ \mathsf{MRSG})$

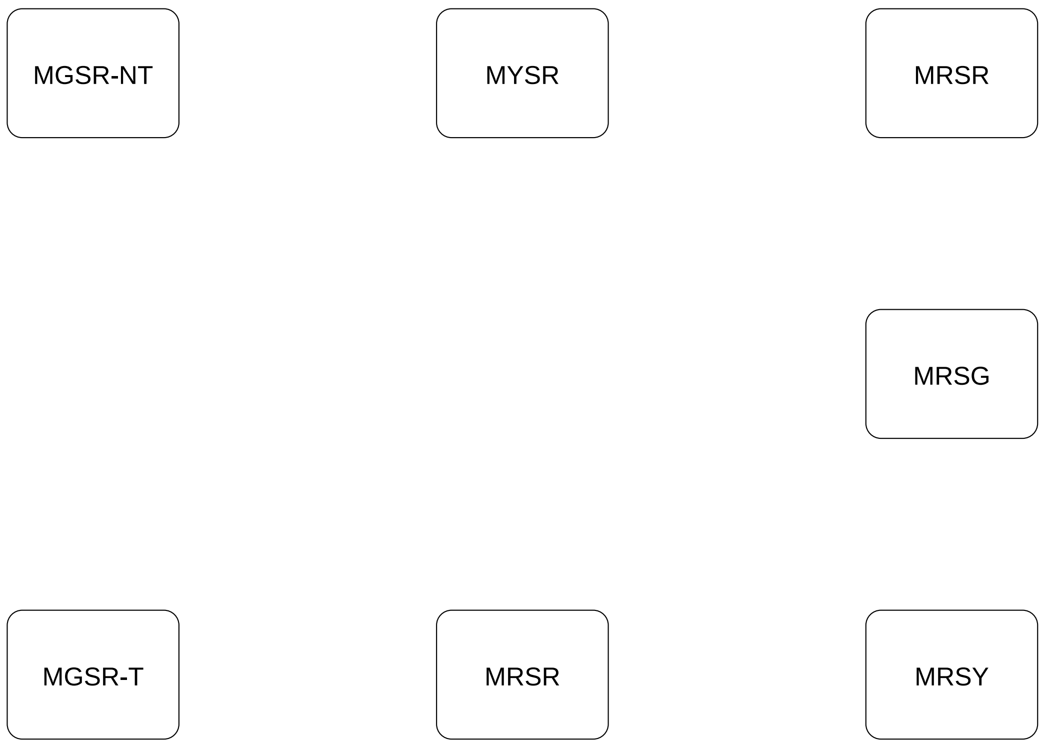
$\Box(\mathsf{MRSG} \Rightarrow (\mathsf{MRSG}\ \mathcal{W}\ (\mathsf{D0C} \vee \mathsf{LTO})))$
$\Box((\mathsf{MRSG} \wedge (\mathsf{D0C} \vee \mathsf{LTO})) \Rightarrow \bigcirc\ \mathsf{MRSY})$

$\Box(\mathsf{MRSY} \Rightarrow (\mathsf{MRSY}\ \mathcal{W}\ (\mathsf{STO})))$
$\Box((\mathsf{MRSY} \wedge \mathsf{STO}) \Rightarrow \bigcirc\ \mathsf{MRSR})$

$\Box(\mathsf{MRSR} \Rightarrow (\mathsf{MRSR}\ \mathcal{W}\ (\mathsf{STO})))$
$\Box((\mathsf{MRSR} \wedge \mathsf{STO}) \Rightarrow \bigcirc\ \mathsf{MGSR\text{-}T})$

$\Box(\mathsf{MGSR\text{-}T} \Rightarrow (\mathsf{MGSR\text{-}T}\ \mathcal{W}\ (\mathsf{LTO})))$
$\Box((\mathsf{MGSR\text{-}T} \wedge \mathsf{LTO}) \Rightarrow \bigcirc\ \mathsf{MGSR\text{-}NT})$

(a) In the skeletal state machine diagram below, draw the finite state machine that is specified by the above linear temporal logic specification. Indicate that MGSR-NT is the starting state, and show all transitions specified by the temporal logic specification. If it turns out that this specification does not specify what is claimed, just carry on, making the state machine equivalent to the temporal logic specification above.

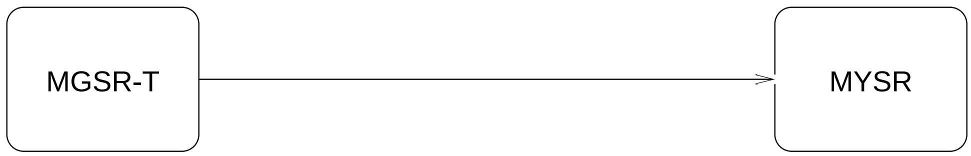


(b) The state machine, as you have drawn it, is not complete. It is necessary after each time out of any duration to reset the external timer to begin the count down to the next two time outs. That is, when the timer is set by the operation ST, at the end of the short time out duration, the event STO occurs. If STO is not detected by any state, then at the end of the long time out duration, the event LTO occurs. There are two ways to have the ST done when it needs to be done: (1) as an action in transitions and (2) as an action to be perform on entry to a state.

i. In your completed state machine diagram, mark with an asterisk ("*"), each transition that needs the action, /ST, in order to achieve the first way to have the ST done when it needs to be done.

ii. In your completed state machine diagram, mark with a plus sign ("+"), each state that needs the entry action, entry/ST, in order to achieve the second way to have the ST done when it needs to be done.

(c) It is possible to add another transition leaving MGSR-T, going directly to MYSR, and bypassing MGSR-NT (but keeping MGSR-NT in the machine as the initial state). **On the state machine fragment given below**, in the standard notation, write the (event, condition, and action) that must be on this transition for the behavior of the whole state machine to be unchanged. Assume here that the ST action is done in the transition.

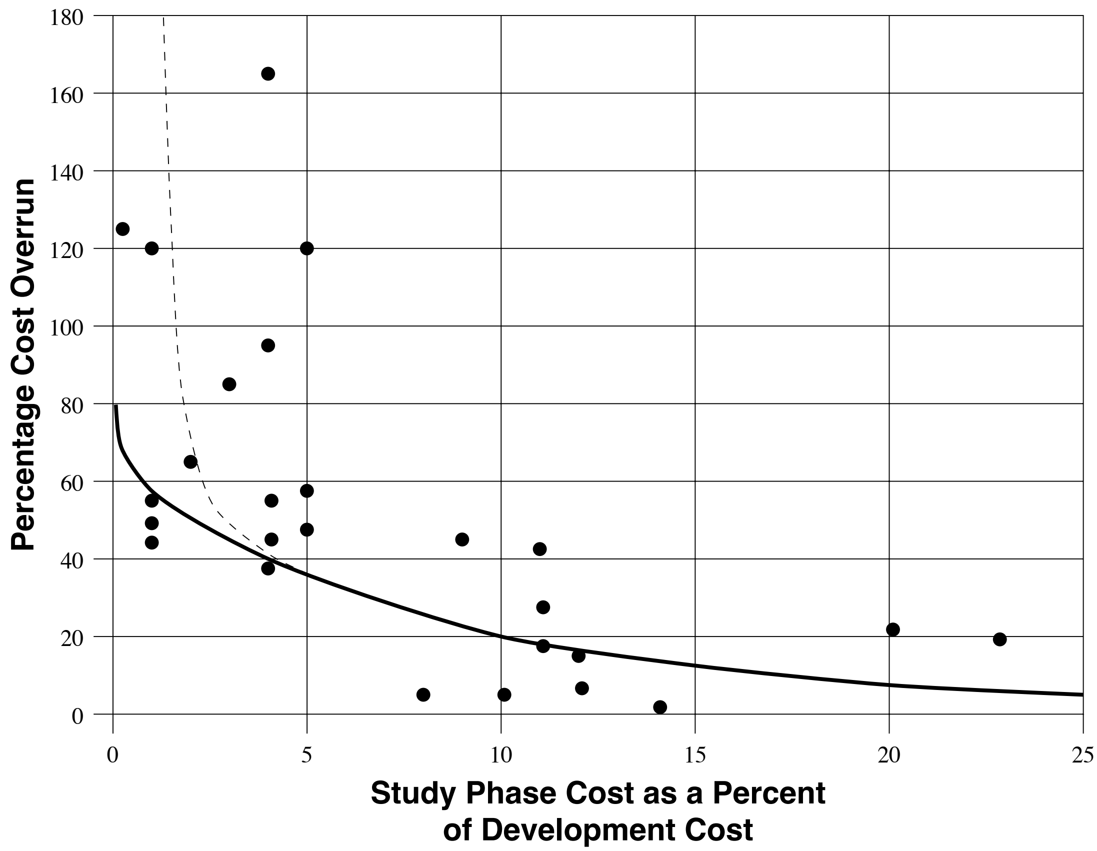


(d) Since each time out, LTO and STO, is guaranteed to happen, some but not all of the occurrences of $\mathcal{W}$ can be changed to $\mathcal{U}$. In the specification above, put an asterisk ("*") above each $\mathcal{W}$ that can be changed to $\mathcal{U}$ and still be correct.

9. **[30 total marks]** The Requirements Iceberg

(a) An E-type CBS is one whose deployment changes the ______________ in which the CBS runs so that the CBS's own ______________ are ______________. Meir Lehmann claims that *any* ______________ that does real ______________ in the ______________ is inherently E-type.

(b) Recall the graph developed by Kevin Forsberg and Harold Mooz relating percentage cost overrun to study phase cost as a percentage of development cost in 25 large NASA projects.



The y-axis is "Percentage Cost Overrun" and not "Total Cost". Therefore, a project with 0% cost overrun might still be ______________ and a project with 100% cost overrun might still be ______________

Give three different reasons, each of which can by itself account for this graph.

i.

ii.

iii.

(c) The lecture titled "Requirements Determination is Unstoppable" reported on a company X, which was serious about doing requirements analysis. However, they were serious also about following schedules and adhering to deadlines. Consequently, on the date that the requirements specifications (RS) for a CBS were to be delivered, whatever RS had been achieved to that date was taken as *the* RS from which the programmers were to implement the CBS, even though the RS was incomplete and incorrect.

When a programmer encounters any incompleteness or incorrectness in the RS, he or she must complete or correct the RS before he or she can continue writing code.

i. What is the ideal way for the programmer to complete or correct the RS?

ii. Give two different reasons why this ideal way does not happen.

A.

B.

iii. When that ideal way is not possible, what often happens instead?

iv. What *is* the difficulty caused by what often happens instead?

v. What is the problem of the lack of benefit of a document to its producers (PotLoBoaDtiP)?

vi. How does PotLoBoaDtiP explain why the delivered SRS for a CBS is not kept up to date by the developers to whom the SRS has been delivered?

(d) We have data that say that 75% – 85% of all defects found during the development, deployment, and use of a CBS can be traced back to an error committed during requirements determination:

- The situation causing the defect was totally ______________ in the CBS's requirements specification.
- The defect was actually ______________ by the CBS's requirements specification.

(e) One famous quotation heard often enough in CBS development is said by a requirement analyst in a position of being able to give programmers orders.

> You start coding while I see what the customer wants.

On one hand, this quotation makes ______________ sense to me, because one ______________ write a piece of code without knowing what the piece of code is supposed to ______________. On the other hand, in some CBS developments, there is an attempt do do *some* development concurrently with requirements determination. They argue that if the programmer has a good idea of some functionality F that will be needed in any case, he or she can start coding F to get a head start. Even if 50% of the code turns out to be wrong, the project is ahead by 50% of the coding of F. What observation cited during class shows that, even if you believe this "50%" figure, this attempt to get a head start is a poor bet?

Therefore, fixing the 50% of the code for F that is wrong will probably cost ______________ than the cost of ____.

In addition, the claim that *only* 50% of the code for F is wrong is suspect because of the observation that (complete the sentence)

(f) In Michael Jackson's famous quotation,

Requirements Engineering is where the informal meets the formal.,

the ______________________________ are the informal and

the ______________________________ and

the ______________________________ are the formal.

The meeting point, during which ______________________________ are specified, even just in natural language, is ____________ because there is no way to write ____________ without knowing what it is supposed to ____________.

(g) One way to classify requirements with respect to the scope of the CBS owning the requirements is into

i. scope ____________ requirements, and

ii. scope ____________ requirements.

For the CBS known as a pocket calculator, give three examples of each kind of requirements.

i. A.

B.

C.

ii. A.

B.

C.