Name_____Student No._____

SOFTWARE ENGINEERING 2A04

DAY CLASS   Dr. PARNAS / Dr. TAYLOR

DURATION OF EXAMINATION: 3 HOURS

MCMASTER UNIVERSITY FINAL EXAMINATION   December, 1998

THIS EXAMINATION PAPER INCLUDES 12 PAGES (INCLUDING COVER SHEET) AND 50 QUESTIONS. YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE. BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

Special Instructions:

Candidates may use calculators, notes, & text books. Answers are to be marked on the exam itself as well as on the OMR answer sheet. In case of discrepancies, the answer sheet is considered to be your answer. Exams and OMR answer sheets are to be collected separately.

OMR EXAMINATION - INSTRUCTIONS

NOTE: IT IS **YOUR** RESPONSIBILITY TO ENSURE THAT THE ANSWER SHEET IS PROPERLY COMPLETED: YOUR EXAMINATION RESULT DEPENDS UPON PROPER ATTENTION TO THESE INSTRUCTIONS.

The scanner, which reads the sheets, senses the shaded areas by their non-reflection of light. A heavy mark must be made, completely filling the circular bubble, with an HB pencil. Marks made with a pen or felt-tip marker will **NOT** be sensed. Erasers must be thorough or the scanner may still sense a mark. Do **NOT** put any unnecessary marks or writing on the sheet.

1. Print your name, student number, course name, section number and the date in the space provided at the top of Side 1 (red side) of the form. Then the sheet **MUST** must be signed in the space marked SIGNATURE (See *).
2. Mark your student number in the space provided on the sheet on Side 1 and fill in the corresponding bubbles underneath.
3. Mark only **ONE** choice from the alternatives (A,B,C,D,E) provided for each question. If there is a True/False question, enter response of 1 (or A) as True, and 2 (or B) as False. The question number is to the left of the bubbles. Make sure that the number of the question of the scan sheet is the same as the question number on the test paper.
4. Pay particular attention to the Marking Directions on the form.
5. Begin answering questions using the first set of bubbles, marked "1". Answer all questions.
6. Please circle the correct answer on the exam itself.
7. Pass **both** the exam sheets and exams in at the end of the exam; they should be collected **separately**.

Name_____Student No._____

The questions below are all based on the lectures, the articles in the "coursepack" and on the book, "The Mythical Man Month". Please read the questions carefully. In many cases all of the alternative answers are true statements, but only one responds to the question. Note too that sometimes we are asking you which statement is **_not_** an appropriate answer.

To answer the next 5 questions, you should complete the following tabular description of a finite state machine. The machine should output a "1" if and only if, the most recent 5 characters (including the current input) were "S E S E M". (Current input is "M", the input before that was "E", ...)

Minimal SESEM Recogniser: Initial State is 1

NS	S	E	M	X		OUT	S	E	M	X
1	2	1	1	1		1				
2	2	3				2			0	0
3	4	1	1			3			0	0
4	2		1			4			0	0
5	4	1		1		5	0	0	1	0

(1)   State 5 is the state of the machine:

A: whenever the last 5 inputs before the current input were S E S E M

B: whenever the last 4 Inputs before the current input were S E S E

C: whenever the last two inputs before the current input were S E

D: whenever the last three inputs before the current input were S E X

E: whenever the last three inputs before the current input were X S E.

(2)   If the most recent two inputs before the current input were M X, the state that you would be in is

A: 1

B: 2

C: 3

D: 4

E: 5

(3)   If the two previous inputs (other than the current input) were S E, the machine could be in

A: state 1

B: state 1 or state 3

C: state 3 or state 5

D: state 4

E: state 2 or state 4.

(4)   If the three most recent inputs (<u>other than the current input</u>) were S E M, the machine could be in

   A: State 1 or state 5
   B: State 1
   C: State 5
   D: State 3
   E: State 4

(5)   If the two most recent inputs other than the current input were S E, the output could be

   A: 1 only
   B: 1 or 0
   C: 0 only
   D: undefined
   E: 2 or 4.

The next 5 questions are based on the pseudo-code program below. Assume that A is a global integer array with subscripts 0:N, and M is a global integer.

(**integer** I; I $\Leftarrow$ N-1; M $\Leftarrow$ A[N];
_**it**_ (I $\geq$ 0 $\rightarrow$ ((($A[I] \leq M$) $\rightarrow$ M $\Leftarrow$ A[I] | ($A[I] > M$) $\rightarrow$ **skip**); I $\Leftarrow$ I - 1; ☞)
    | I $<$ 0 $\rightarrow$ ●)
_**ti**_)

(6)   What is a monotonically decreasing quantity that can be used to confirm termination of the loop?

   A: I + 1
   B: A[I]
   C: N - I
   D: I - A[I]
   E: M - I.

(7)   Which of the following predicates is invariant in the loop

   A: $I > 0$
   B: ($\forall$ j, $I \leq j \leq N \Rightarrow M \leq A[j]$)
   C: $A[I] \leq M$
   D: ($\forall$ j, $I < j \leq N \Rightarrow M \leq A[j]$)
   E: None of the above.

(8)   The program above terminates with M equal to:

   A: the largest value in the array
   B: A[N] - if A[N] is larger than any other element of the array.
   C: **skip**
   D: the smallest value in the array
   E: None of the above

(9)   If N is initially 0,

    A: The array has only one element.

    B: The program will not terminate normally but would fail when accessing an array element that does not exist.

    C: The program would loop forever.

    D: The program would terminate with $I = 0$.

    E: None of the above

(10) The program will

    A: Operate normally only if $N \geq 0$.

    B: Operate normally if $N < 0$.

    C: Execute **skip** repeatedly if $N = 0$.

    D: Operate normally in all cases

    E: Never terminate

The next 5 questions are based on the pseudo-code program below. Assume that L, N, and S are global integers, initially $N \geq 0$, and A is a global integer array with subscripts 0:N.

(**integer** I; $I \Leftarrow N$; $L \Leftarrow 0$; $S \Leftarrow L$; *it* seespec; ($I > 0 \rightarrow (I \Leftarrow I - 1;$ ☛) | $I \leq 0 \rightarrow$ ●) *ti*)

seespec is a program that satisfies the following specification

	A[I] > 3	A[I] = 3	A[I] < 3
L' =	'L + 1	'L	'L
S' =	'S	'S	'S + 1

(11) What is a monotonically decreasing quantity that can be used to confirm termination of the loop?

    A: I

    B: A[I]

    C: N - I

    D: I - A[I]

    E: L - I.

(12) Which of the following best describes the function of seespec

    A: Increment L if A[I] is greater than 3.

    B: Decrement L if A[I] is greater than 3.

    C: Do nothing if A[I] = 3

    D: Decrement S if A[I] is less than 3

    E: Increment L if A[I] is greater than 3 and increment S if A[I] is less than 3.

(13) Which of the following is invariant in the loop.

    A: $L \leq S$

    B: $S \leq L$

    C: $L \leq I$

    D: $L > N$

    E: None of the above

(14) Which of the following is true if $N < 0$?

    A: The program seespec is never used.

    B: The program seespec will abort or give strange results because of an improper array access.

    C: The program seespec will be called $|N| + 1$ times.

    D: The program seespec will be called infinitely many times.

    E: None of the above

(15) In this question, if "E" is true, you must chose "E".
If all elements of the array A have the value 3:

    A: The final value of L will be 0.

    B: The final value of S will be 0.

    C: seespec will be called $N + 1$ times.

    D: The final value of I will be zero.

    E: All of the above.

(16) During the design of the OFP (Onboard Flight Program) for the A-7 aircraft, described in the assigned readings, it was proposed that there be two major modules. One module would be responsible for calculating the current position and velocity of the aircraft; the other module would be responsible for calculating weapon impact points. This was **_not_** done. The reason was:

    A: it was important that the weapon delivery work even when the navigation fails.

    B: in military aeroplanes, navigation without weapon delivery capability is useless.

    C: these two modules were too large to be top-level modules in the hierarchy.

    D: some calculations were used for both navigation and weapon delivery and it would be unclear which module should contain those.

    E: weapon delivery functions should never be used unless navigation is working.

(17) The "mythical man month effect", as discussed in the assigned readings means:

    A: we should not assume that men and women write software differently.

    B: because different people write code at different speeds, we cannot measure the complexity of a product in man-months.

    C: doubling the number of programmers, may not cut the development time in half.

    D: it takes much more time to write code than we think it will.

    E: it is important to include testing time in your cost estimates.

(18) Which of the following statements about modules was **_not_** stressed in this course?

  A: It is important that modules can be compiled separately.
  B: It is important that modules can be tested separately.
  C: It is important that modules can be written separately.
  D: It is important that modules not include programs A and B such that A uses B and B uses A.
  E: It is important that modules have interfaces that are unlikely to change.

(19) Software seems to age. Which of the following is **_not_** one of the reasons given in your reading assignments?

  A: Requirements change as markets develop.
  B: Alterations to software often change the modular structure.
  C: Documentation is not kept up to date when changes are made.
  D: Magnetic media degrade.
  E: Changes may not be consistent with the original design concepts.

(20) Which of the following statements is a true statement about module structure?

  A: It is important to describe the modules in the order of their execution.
  B: It is important to describe the modules in alphabetic order.
  C: It is important to put the most difficult modules at the bottom of the structure.
  D: Two programs should be in the same module if they have shared design decisions that are likely to change.
  E: Two programs should be in the same module whenever you would not use one without the other being present.

(21) Which of the following statements is **_not_** true of the second modularisation of the KWIC index problem given in "On the Criteria to be used in Decomposing Systems into Modules"?

  A: It would be possible to implement the Alphabetiser module in such a way that ITH is a very simple program.
  B: It would be possible to implement the Alphabetizer module in such a way that ALPH is a very simple program.
  C: All superficial changes in the format of the input could be accommodated by changing only the INPUT module.
  D: It would be possible to build the system without using DELINE and DELWRD from the Line Storage module.
  E: To save space, one could use a symbol table module within the Line Storage module; this would require a change to at least one other module.

(22) Which of the following positions *is* taken in the readings on modularisation?

A: If two programs are always used at the same points in other programs, they should be part of the same module.

B: If A and B are programs, and A is always called immediately after B is called, A and B should be in the same module.

C: If two programs contain direct references to the same array, they should probably be part of the same module.

D: Programs that change the values in a data structure should be together in one module while those that return values derived from the module's data should be in another module.

E: It is important to make sure that all programmers know the formats used in files or other "persistent" data structures.

(23) Which of the following is *not* one of the reasons to design programs that can be easily extended and contracted?

A: We want to have a subset that we can deliver to a customer even if the project misses its deadlines.

B: We want to be able to squeeze a system into limited memory if it will help us to increase the market for the program.

C: We want it to be easy to add features to keep ahead of the competition

D: We want to help hardware manufacturers to convince customers to buy newer, bigger and faster computers.

E: Programs designed for extension and contraction are easier to test during development.

(24) Building software as layers of virtual machines is a good idea because:

A: virtual machines are cheaper than real ones.

B: virtual machines are more powerful than real ones.

C: virtual machines are more convenient than real ones because they abstract from certain details.

D: the upper-level machines exercise firm control over the lower-level machines

E: middle layers can be removed just by renumbering the levels.

(25) If we have two programs A and B, which of the following is *not* required if A is to be allowed to use B?

A: B should not be more complex because it cannot use A.

B: A should be simplified because it uses B.

C: There should be a useful subset with B and without A.

D: There should be no useful subsets having A and not B.

E: B should be more useful than A.

(26) In the major example in "Designing Software for Extension and Contraction", OUTADSEL is a level above INAD because:

    A: outputs are more important than inputs in requirements analysis.

    B: OUTADSEL uses OUTAD and INAD uses nothing.

    C: you cannot output data unless you have read it in first.

    D: the address output module is placed at a higher level than address input.

    E: OUTAD has to use the address storage module.

(27) A software structure can be termed hierarchical if and only if:

    A: no component is a part of two components.

    B: each component has resources that can be allocated to its subcomponents.

    C: the software structure is represented by a directed graph that has no loops.

    D: the software structure can be represented by a directed graph in which each node has at most one arc coming into it.

    E: every program has been assigned a definite level of difficulty.

(28) The "gives work to" relation should be hierarchical because:

    A: if a program gives work to itself, it will never terminate.

    B: programs that give work to themselves may end up in a deadlock.

    C: if programs don't give work to themselves (even indirectly), deadlocks are not possible.

    D: It is unwise to create work for your boss.

    E: if there are no loops in the "gives work to" hierarchy, it is easier to determine whether or not a system may deadlock.

(29) Which of the following is _**not**_ a good guideline for module design?

    A: It should be possible to change one module without changing others.

    B: Modules should consist of one program with one entrance and one exit.

    C: The most likely changes should not affect the interface of the module.

    D: Design decisions that might change should be "known" to only one module.

    E: None of the above

(30) Which of the following statements is true?

    A: The module guide reveals the secrets of each module.

    B: The module guide sketches the interfaces for each module.

    C: If you have a module guide, you don't need module specifications.

    D: The module guide is intended to help a programmer to find out which modules will be affected by a proposed change.

    E: The module guide contains the specifications for all of the modules.

(31) In the A-7 structure described in your notes:

    A: the Hardware Hiding Module hides what the hardware will be used for.

    B: the Behaviour Hiding Module hides the behaviour and interfaces of special purpose devices.

    C: the Software Decision Module hides the algorithms used in physical models.

    D: the Behaviour Hiding Module hides the shape of the HUD symbols.

    E: the Hardware Hiding Module hides the meaning of the HUD symbols.

(32) In the A-7 software, the Extended Computer Module would have to be changed if:

    A: the computer was extended by adding a floating point co-processor.

    B: the computer was replaced by a clone with more robust circuitry.

    C: it was decided to use high precision variable to represent altitude instead of a normal precision variable.

    D: it was decided to replace the Air Data Computer.

    E: a colour display replaced the current HUD.

(33) In the A-7 software, mode definition tables are secrets of the Shared Services Module because:

    A: mode definitions affect more than one function driver.

    B: mode transition programs are very frequently used.

    C: the mode definitions must be known to both pilots and programmers.

    D: mode definitions affect more than one device interface module.

    E: the same definitions must be used in many aircraft or the pilots will be confused.

(34) In the A-7 software, the Physical Model module:

    A: calculates the sine, cosine and tangent of latitude and longitude.

    B: calculates altitude given the air pressure.

    C: simulates wind tunnel behaviour and is used in designing the wings.

    D: would have to be changed if the representation of latitude and longitude are changed.

    E: would have to be changed if the Projected Map Display Set was to show the aircraft position at the top of the display instead of at the centre or the bottom of the display.

(35) In the A-7 software, if all of the HUD (Head Up Display) symbols were required to flash when the pilot should take some action such as "pulling up":

    A: it would affect the HUD Device Interface Module.

    B: it would affect the HUD Function Driver Module.

    C: it would affect the Device Interface Module for the Angle of Attack sensor.

    D: it would require changes in the Aircraft Motion Module.

    E: it would require changes in the Human Factors Module.

(36) In the THE Multiprogramming System:

A: deadlock was prevented by the use of semaphores at level 1.

B: semaphores could be created when needed or deleted when not needed.

C: semaphores were used for mutual exclusion.

D: the semaphores could be paged in and out like all other data.

E: level 2 could not be used if the drum had broken down.

(37) When planning a review for a Device Interface Module (DIM) specification, *programming experts* should __not__ be expected to know:

A: if the interface is based on assumptions that are valid for all possible replacement devices.

B: if the assumption list for the device are sufficient to allow the device to be used properly.

C: if the assumptions contain enough information to be sure that the access programs can be implemented.

D: if the proposed interface is consistent with programming language conventions.

E: if the proposed interface can be implemented for the current device.

(38) In "The Mythical Man Month", Fred Brooks states that:

A: more projects have gone awry for lack of calendar time than all for other causes combined.

B: you must be unusually optimistic to be a good programmer.

C: when you partition tasks, do so in such a way that there need be no information shared between the programmers.

D: dividing a program into parts to be constructed separately will increase costs.

E: software is best written by a lonely individual without interference.

(39) In the early part of his book, Brooks suggests that you should plan to build a prototype and then discard it and start over. Later he concludes that this advice (given 20 years earlier) was wrong. The reason that he gives is:

A: your boss or customer will not let you throw it away.

B: if you build a system with the intent of discarding it, you learn how to build systems that should be discarded.

C: it is better to use an incremental building technique with each new extension tested before you add to it.

D: the second system might be worse than the first.

E: none of the above.

(40) When well designed software projects run out of time, the best thing to do is:

A: Ask for a small extension in time.

B: Skip testing.

C: Hire someone new to write the documentation.

D: Eliminate programs that are low in the uses hierarchy from the first release.

E: Eliminate programs that are high in the uses hierarchy from the first release.

(41) The "ideal" software design process is one in which the design is derived from a clear statement of requirements. Which of the following is **_not_** one of the reasons for the fact that this process is seldom followed in real software developments?

  A: It is difficult to find out all of the requirements at the start of a project.

  B: The developers think the requirements are obvious and don't write them down.

  C: People want to reuse or extend previously written software that may not be ideal for the new requirements.

  D: People find it easier to write code than to describe requirements precisely.

  E: None of the above.

(42) Which of the following would be a secret of a hardware hiding module designed to make it easier to replace terminals in a computer system with newer hardware.

  A: Whether the messages are displayed in English or French

  B: The space between the buttons on the keyboard.

  C: Whether or not windows are allowed to overlap

  D: How to display an "8" at a given point on the screen.

  E: Where to display an "8" on the screen.

(43) Which of the following would Fred Brooks now agree with:

  A: Conceptual integrity is *the* most important ingredient in system design.

  B: To increase the function of a system, one must increase its complexity.

  C: A conceptually integrated system will be slower to build but faster to test.

  D: Interface design and implementation can proceed in parallel.

  E: It is impossible to include testing time in your planning estimates.

(44) The interface specification of a module designed to store tree structured data should

  A: not reveal that there is a loop-free "child of" relationship between the elements.

  B: not reveal that there is only one node that is not a child of another

  C: not reveal the limits on the number of elements that can be stored

  D: not reveal the types of the data elements

  E: not reveal how the links between data elements are stored by the module.

(45) Which of the following statements is a true statement about module structure?

  A: One must think carefully about whether or not the "part of" relationship should be hierarchical.

  B: It is useful to have an alphabetical list of module names.

  C: It is important to put the most difficult modules at the bottom of the structure.

  D: Two programs should be in the same module if they can be implemented using similar data structures.

  E: Two programs A, B should be in the same module whenever you would not use A without B being present.

(46) In this question, if E is true, choose E.
Which of the following modules should include code to calculate sine and cosine functions?

A: the Function Driver modules.
B: the Device Interface Modules that use angle representations.
C: the Shared Services submodule of the Behaviour Hiding Module.
D: the Software Decision modules.
E: all of the above.

(47) In this question, if D is true, choose D.
A well-structured telephone system offers its operators a choice of messages in English and French. Because of foreign sales, they want to add Dutch. This will affect:

A: the Device Interface Module for the operator terminals.
B: the Function Driver Module.for the operator terminals.
C: the interprocess communications module.
D: all of the above.
E: none of the above.

(48) If we want to demonstrate that a program's reliability is probably at least 0.999, we need to:

A: perform around 1000 randomly selected tests without encountering any failures.
B: perform around 5000 randomly selected tests without encountering any failures.
C: perform around 5000 tests without encountering more than five failures.
D: allow the program to run unattended for 48 hours.
E: None of the above is enough.

(49) In "The Mythical Man Month", Fred Brooks states that:

A: Dividing a program into modules will increase costs.
B: You must be unusually pessimistic to be a good tester.
C: The "architect" should specify programs and then instruct the programmer about how to implement the specification.
D: One needs both a formal definition of a design for precision and a prose definition for comprehensibility.
E: It is impossible to accurately estimate the time and effort that a software project will require.

(50) When he says "plan to throw one away" Brooks means that:

A: you will always fail the first design review; count on it.
B: you can learn a lot from building a prototype; you will then do things better.
C: you won't get it right the first time so you might as well not try.
D: most of the documentation will not be used and can be discarded.
E: None of the above.

THE END