Part 1
Computer Basics
Study Guide

Coverage:

1. Von Neumann architecture – need to know what it is and why it is important. Also be familiar with the concept of a computer consisting a hierarchy of virtual machines.
2. Different levels in a computer system & their significance.
3. Organization of a simple Processor and its functioning
4. Byte ordering (Little Endian, Big Endian)
5. Error Correcting codes.
6. Digital Logic Level (Combinational & Sequential , All major concepts)
7. Memory Hierarchies
8. Memory characteristics – virtual memory, cache memory, RAM and ROM characteristics, memory addressing , data transfer from memory to CPU
9. Internal representation – character data, integers, floating point, boolean, instruction set
10. Components of a working computer system
11. Principle of Programming languages: machine code, assembly language, and high level programming languages
12. Concept of application software and system software
13. Concept of operating systems and language translators
14. Interrupts- What is an interrupt, what causes one to occur
15. History of computing
16. Computer ethics
17. Phases of software development

Computer Organization Questions:

Levels in a Computer System,Parts of a Microprocessor , Conventions, Error Correcting codes, Assembly Language Questions 1-4

1. The fundamental conceptual unit in a computer is:
   a. CPU
   b. Hard Drive
   c. Operating System
   d. Transistor

2. In a 8086/8088 Microprocessor, the unit responsible for getting the instructions from memory and loading in the Queue is.
   a. Execution Unit
   b. Registers
   c. Stack
   d. **Bus Interface Unit**

3. When you transfer the record from a Big Endian system to a Little Endian system over the network in order to get the original value, you must:
    a. reverse the byte within a word
    b. reverse the bytes in an integer
    c. reverse the characters in a word
    d. there is no simple solution

4. To transmit data bits 1011, the correct even parity seven bit Hamming Code is
    a. 0101101
    b. 1010101
    c. 1100111
    d. 0110111

Digital Logic, Addressing Modes etc. (Questions 5 to 8)

5. In a Sequential Logic system, the out put remains even after the input is removed
    a. True
    b. False

6. The cause of propagation delay is the time it takes a pulse to get through a logic device
    a. True
    b. False

7. If CS= 24F6 and IP =634A, the physical address is
    a. 24F6: 634A
    b. 34F5F
    c. **2B2AA**
    d. 24F60

8. The instruction MOV CL, [BX][DI]+8 represents the following addressing mode
    a.based relative
    b.based indexed
    c.indexed relative
    d.register indirect

Secondary Storage and Disk Drives – Questions 9-11

9. The amount of time required to read a block of data from a disk into memory is composed of seek time, rotational latency, and transfer time. Rotational latency refers to
   a. the time it takes for the platter to make a full rotation
   b. the time it takes for the read-write head to move into position over the appropriate track
   c. the time it takes for the platter to rotate the correct sector under the head
   d. none of the above

10. If a magnetic disc has 100 cylinders, each containing 10 tracks of 10 sectors, and each sector can contain 128 bytes, what is the maximum capacity of the disk in bytes?

    a. 128,000
    b. 12,800,000
    c. 12,800
    d. 1,280,000

11. According to the specifications of a particular hard disk a seek takes 3 msecs (thousandths of a second) between adjacent tracks. If the disk has 100 cylinders how long will it take for the head to move from the innermost cylinder to the outermost cylinder.
    a. 30 microseconds
    b. 300 msecs
    c. 3000 msecs
    d. 3 microseconds

Parameters: Any questions related to the understanding of disk structure and operation

Primary memory hardware – Questions 12-16

Primary memory is usually implemented using both ROM (Read only memory) and RAM (Random Access memory) chips. Both of these chips provide random access to memory words. The difference between them is that ROM cannot be changed during the normal actions of a program, hence its name, read only memory, while RAM memory may be both read and written. The other difference between them is that RAM memory loses its information when the power is turned off to the chip, but ROM memory does not. For this reason ROM memory is considered non-volatile.

12. What characteristic of RAM memory makes it not suitable for permanent storage?
    a. too slow
    b. unreliable
    c. it is volatile
    d. too bulky

13. Part of the operating system is usually stored in ROM so that it can be used to boot up the computer. ROM is used rather than RAM because
    a. ROM chips are faster than RAM
    b. **ROM chips are not volatile**
    c. ROM chips are cheaper than RAM chips
    d. none of the above

14. A given memory chip has 12 address pins and 4 data pins. It has the following number of locations.
    a. 2^{4}
    b. $\mathbf{2^{12}}$
    c. 2^{48}
    d. 2^{16}

15. RAM is called DRAM(Dynamic RAM) when
    a. a.it is always moving around data
    b. b.it requires periodic refreshing
    c. it can do several things simultaneously
    d. none of the above

16. Which of the following is Non-Volatile memory?
    a. EEPROM
    b. SRAM
    c. DRAM
    d. None of the above

Parameters for 5-7: Other characteristics of RAM and ROM chips. The definition of random access.

<u>Internal Representation – Questions 17-22</u>

Computers store all data and instructions as 0's and 1's. For this reason the numbers and characters we use to write instructions and data must be translated or encoded as 0's and 1's before they can be used in a computer. This encoding is usually called the internal representation of the character or number. In virtually all personal computers, the 7 bit ASCII code is used to internally represent characters. Any positive whole number can be transformed into its binary equivalent, but two's complement notation is generally used to internally represent integers because of it efficiently represents both positive and negative numbers. Real numbers (fractions) are usually encoded using floating point notation.

17. Two’s complement notation is frequently used for internal representation of
    a. fractions
    b. integers
    c. True and False values
    d. floating point numbers

18. If the ASCII code for A is 1000001, B is 1000010, and C is 1000011 then the string 100001110000011000010 represents:
    a. CAB
    b. BAC
    c. CCB
    d. ABC

19. The two’s complement representation of –10 is:
    a. 11110110
    b. 11011001
    c. 00001010
    d. 11111100

20. The binary representation of 15 is:
    a. 01010
    b. 01111
    c. 10011
    d. 00101

21. Floating point representation is used to store
    a. Boolean values
    b. whole numbers
    c. real numbers
    d. integers

22. Binary numbers can be used to represent
    a.Integers only
    b.Fractions only
    c.Both fractions and integers
    d. both fractions and integers.

Parameters: The numbers or characters to be converted to their internal representation. The order of conversion, i.e. the question may be to convert from two’s complement to an integer.

Computer Buses and performance – Questions 23 - 25

A bus is a set of wires connecting computer components. A computer may have several buses, e.g. a system bus, an internal bus, and special purpose local buses. All communication between the various components takes place over one of these buses. For example, data transfer between the CPU and memory normally occurs on the system bus, while movement of data between registers and the ALU takes place on a bus internal to the microprocessor chip. The speed at which data can be transferred is dependent on the number of data lines in the bus and, in the case of synchronous buses, the clock speed of the bus. The transfer rate or bandwidth of a particular system bus can be calculated from the number of cycles required for transfer, the length of the cycle and the number of data lines. For example, if a bus has 8 data lines, requires 4 cycles to transfer data, and each cycle is 250 nsecs then the bandwidth of the bus is 1 byte per 1000 nsecs. This is equivalent to 1 byte per microsecond or 1 Megabyte /second.

23. In order to execute a program instructions must be transferred from memory along a bus to the CPU. If the bus has 8 data lines, at most one 8 bit byte can be transferred at a time. How many memory access would be needed in this case to transfer a 32 bit instruction from memory to the CPU.
    a. 1
    b. 2
    c. 3
    d. 4

24. Suppose that a bus has 16 data lines and requires 4 cycles of 250 nsecs each to transfer data. The bandwidth of this bus would be 2 Megabytes/sec. If the cycle time of the bus was reduced to 125 nsecs and the number of cycles required for transfer stayed the same what would the bandwidth of the bus?
    a. 1 Megabyte/sec
    b. 4 Megabytes/sec
    c. 8 Megabytes/sec
    d. 2 Megabytes/sec

25. Any computer must at least consist of

    a. Data bus
    b. Address Bus
    c. Control Bus
    d. all of the above

Parameters: different types of buses, length of cycles, number of cycles, number of data lines.

Memory Structure – Questions 26 – 30

Computer memories can vary in the number of addressable units they contain and the size of these addressable units. In many computers the smallest addressable unit is an 8 bit byte. This is not a universal characteristic, however. In some cases the smallest addressable unit is a 4 byte word (32 bits), and there have been computers whose memories were composed of words of size 60. The number of addressable units in a memory determines how many bits are required for a memory address. For example, if a memory contains 64K bytes and a byte is the smallest addressable unit then 16 bits are required for a memory address. On the other hand, if the smallest addressable unit in this memory is a 32 bit word, then 14 bits are required for a memory address.

26. A computer's memory is composed of 8K words of 32 bits each. How many bits are required for memory address if the smallest addressable memory unit is a word?
    a. 13
    b. 8
    c. 10
    d. 6

27. A computer's memory is composed of 4K words of 32 bits each. How many total bits in memory?
    a. 12800
    b. 1280000
    c. 1310720
    d. 131072

28. A computer's memory is composed of 8K words of 32 bits each, and a byte is 8 bits. How many bytes does this memory contain?
    a. 8K
    b. 32K
    c. 16K
    d. 4K

29. A computer's memory is composed of 8K words of 32 bits each, and the smallest addressable memory unit is an 8 bit byte. How many bits will be required for the memory address?
    a. 12
    b. 15
    c. 13
    d. 10

30. A “word” is the natural unit of organization of memory. Different computer types may have different word lengths (in bits) . **True** or False?

Parameters: Different memory sizes and different addressable unit sizes

<u>Memory Hierarchy – Questions 31 – 38</u>

Computer memory can be viewed as a hierarchy. At the top of the hierarchy is the fastest memory. This memory is usually the smallest and most expensive. Registers and cache memory fall into this category. The next level in the hierarchy is primary memory. This memory is larger than cache memory and is cheaper and slower. At the bottom of the hierarchy is disk storage, which is often called secondary memory. Disk storage is the cheapest and slowest memory. The more immediate the need for the data, the higher in the hierarchy it is stored. For example, data needed for an instruction that is being executed in the CPU is stored in the registers, while data needed for the next few instructions is stored in the cache. Instructions and data which are not immediately required are stored in primary memory or on the disk.

31. Cache memory refers to
   a. cheap memory that can be plugged into the mother board to expand main memory
   b. fast memory present on the processor chip that is used to store recently accessed data
   c. a reserved portion of main memory used to save important data
   d. a special area of memory on the chip that is used to save frequently used constants

32. Registers contain data and instructions needed by the CPU. **True** or False?

33. A computer that is advertised as having a 96K byte DRAM memory and a 2.1 Gigabyte hard drive has

   a. **96 K bytes of primary memory and 2.1 Gigabytes of secondary memory**
   b. 2.1 Gigabytes of primary memory and 96K bytes of secondary memory
   c. 96 bytes of cache, 2.1 gigabytes of primary memory
   d. 96K bytes of cache, 96 K bytes of primary memory , and 2.1 Gigabytes of secondary memory

34. A memory management technique used to improve computer performance is

   a. selecting memory chips based on their cost
   b. storing as much data as possible on disk
   c. using the cache to store data that will most likely be needed soon
   d. preventing data from being moved from the cache to primary memory

Parameters: Any question about the function of caches, registers, primary and secondary memory.

35. The fetch-decode-execute cycle refers to the process by which data is read from the hard drive and stored in memory. True or **False**?

36. Interrupts can be generated in response to
    a. detected program errors such as arithmetic overflow or division by zero
    b. detected hardware faults
    c. Input/Output activities
    d. Internal timers
    e. b, c, and d
    f. a, b, c, and d

37. Virtually all computer designs are based on the von Neumann architecture. A high level view of this architecture has the following three components:
    a. Buses, memory, input/output controllers
    b. Hard disks, floppy disks, and the CPU
    c. memory, the CPU, and printers
    d. memory, input/output modules, and the CPU

38. Which of the following programming languages has an instruction set closest to the machine language of a computer?
    a. BASIC
    b. Fortran
    c. Assembly Language
    d. C++

History of Computing – Question 39 – 40

39. The first person who published paper on using computers to perform tasks other than computations is
    a. Charles Babbage
    b. Lady Lovelace
    c. Alan Turing
    d. Konrad Zuse

40. What was the name of the government funded computer used during World War II to compute firing tables?

    a. VAX computer
    b. IBM computer
    c. Colossus computer
    d. ENIAC computer

Parameters: Any questions related to the evolution of computer technology

General questions about Operating Systems – Questions 41 to 43

41. Which of the following started out as separate program from operating systems but usually is included as part of the OS later on?

    a. **Text Editor**
    b. Command processor
    c. Resource allocator
    d. Dispatcher

42. An operating system that allows several processors to perform computation as the same time is call

    a. Single program
    b. Multitasking
    c. **Multiprocessing**
    d. Real time processing

43. The kind of interface with icons and menu bars for user to point at with mouse instead of entering commands for operating system to perform certain tasks is called

    a. **GUI**
    b. Command line interface
    c. User friendly programming
    d. None of the above

Parameters: Other general questions regarding the functions, components and operations of an operating system may be asked.

Computer Ethics – Question 44 to 46

44. he prime targets of software pirates are
    a. programs written for mainframes
    b. **games and application programs for microcomputers**
    c. programs in the public domain
    d. none of the above

45. Considering the impacts computer technology has on modern society, which of the following is negative?

    a. People can communicate with each other without geographical limitations
    b. Stock brokers do not need to be in a central place for trading
    c. **More people totally rely on computers when doing their works**
    d. Information can reach different area and people at light speed

46. A student retrieve a copy of a program assignment from the recycle bin in the computer lab and use the code to complete his own program assignment. His action is considered

    a. **legal but unethical**
    b. illegal and unethical
    c. legal and ethical
    d. illegal but ethical

Parameters: Any questions related to the topic may be asked.

Part II.
Computer Programming
Study Guide

Coverage:

Topics covered in Computer Programming I (CSC170) and Computer Programming II (CSC260). Including:

1. Problem solving
2. Variables, constants, standard and programmer defined data types
3. Operations and operators that manipulating data of various types
4. Structure programming
5. Control structures
6. Recursions and programming style
7. Simple data structures such as Arrays, Records, and Sets
8. Manipulation of Text Files

---

1. Suppose we have

```
int s[6][6];
int j, k;

for ( j = 0; j < 6; j++ )
        for ( k = 0; k < 6; k++ )
                s[j][k] = ( j+k) % 6
```

What is the value of s[s[5][3], s[1][3]]?

a. 1   c. 3   e. 5
b. 2   d. 4

Parameters:
The parameters are the size of the array, the loop limits and the type of loop

2. What is the output of the following program?

```
#include <iostream>
void tttttt ( int & a, int & b );
int main()
{
        int m = 10, n = 100;
        tttttt( m, n );
        cout << m << "   " << n << endl;
        return 1;
}
void tttttt( int & a, int & b )
{
        int t = a;
        a = b;
        b = t;
{
```

a. 10 10 **c. 100 10** e. none of the above

b. 10 100 d. 100 100

Parameters:

The program.

3.

```
int a = 10;
while ( a >= 0 )
        a = a - 3;
cout << a;
```

What is the output?

a. -3 c. -1 e. 10

b. -2 d. 0

Parameter:

<u>The type of loop and the loop body</u>

4. Given A = FALSE; B= TRUE . Which one of the following is TRUE?

a. B implies (A and not B)

b. not (A or B)

c. not (not A and B)

d. not (A or not B)

e. B implies A

Parameters:
The values for A and B

5.

```
int F ( int a, int b )
{
        return ( a + b + 1 );
}
```

What is the value of f (f(10,7), f(5,3))?

a. 27	c. 34	e. 36
b. 33	d. 35	

Parameters:
The function body and the number of input parameters

6. The C++ assignment for the algebraic equation:

$$X = \frac{A + B}{C - D}$$

a. X = A + B / C - D
b. X = A + B / (C - D)
c. X = (A + B) / (C - D)
d. X = (A + B) / C - D
e. X = A + (B / C) - D

Parameters:
The arithmetic expression

7. Suppose
variables, a and b are global integer variables (declared before main() and the prototype for function zzz and

```
int a, b;
void zzz( int & t, int d );

int main()
{
        a = 10;
        b = 20;
        zzz( b, a );
        cout << a << " + " << b << endl;
        return 1;
}

void zzz( int & c, int d )
{
        int b = 100;
        c = b + d;
        a = 6;
}
```

What is the value of the output?

a. 26
b. 106
c. 116
d. 20

Parameters:
The program

8.
```
for ( int j = 1; j <= N; j++ )
        for ( int k = 1; k <= j; k++ )
                sum = sum + 1;
```

How many times is the statement " sum = sum + 1 " in the above program segment executed at the end of the loops?

a. n * n times
b. n * n - n times
c. n / 2 * (n + 1) times
d. ((n*n)/2) - (n/2) times

Parameters:
<u>The type of loop</u>

9. For an applications program, program documentation external to the program (as opposed to comments embedded in the code) should

a. be brief, be written in telegraphic style, use ample diagrams, and be designed for use by the original programmer only
b. be comprehensive, including information for users as well as technical information, to facilitate future revisions
c. be discarded once the program is finished, in order to protect the copyrighted. be written only if required by the job supervisor
e. never be written by the original programmer

10. When deciding on a particular algorithm to use in a program, which of the following should be taken into consideration?

I. The speed of implementation for the algorithm.
II. The space requirements of the algorithm.
III. The ease with which the logic of the algorithm can be understood.

a. II only b. III only
c. I and II only d. II and III only **e. I, II, and III**

11. Consider the following poorly formatted C++ program fragment.

```
if ( a == 7 ) if ( c == 6 ) { c = 9; d = 9; } else { t = 10; if ( c == 6 ) c = 5 } else p = 9;
```

if a = 7 and c = 6 before the code fragment is executed, which of the following indicates the values of a, c, d, p, and t after the fragment is executed? (An undetermined value is indicated by a question mark.)

	a	c	d	p	t
a.	**7**	**9**	**9**	**?**	**?**
b.	7	5	?	?	10
c.	7	6	?	?	?
d.	7	5	9	?	10
e.	7	6	?	9	?

Parameter:
<u>The program</u>

12.

```
int result ( int m, int n )
{
        int temp = 1;
        for ( int j = 1; j < n; j++ )
                temp = temp * m;
        return temp
}
```

Of the following, which best indicates the task performed by the function above if only positive parameters are passed to the function?

a. It approximates the mth root of n.
b. It approximates the nth root of m.
c. It computes the mth power of n.
d. It computes the nth power of m.
e. It computes m*n

Parameter:
The function

13 A program uses a plotting board in the form of a Cartesian plane with the center of the board at (0,0). A pen is suspended over the board and can be controlled by the following instructions:

RP	-- Raise pen
LP	-- Lower pen
MP(x,y)	-- Move pen to point (x,y)

If the pen is up when MP(x,y) is executed, it will stay up; otherwise, it will draw a line segment from its current position to (x,y). Which of the following sequences of instructions will cause the capital letter T, and nothing else, to be drawn?

a. RP, MP(0,0), LP, MP(0,20), MP(-5,20), MP(5,20)
b. RP, MP(0,0), MP(0,20), MP(-5,20), MP(5,20)
c. RP, MP(0,0), LP, MP(0,20), RP, MP(-5,20), MP(5,20)
d. RP, MP(0,0), LP, MP(0,20), RP, MP(-100,20), MP(100,20)
e. RP, MP(0,0), LP, MP(-5,20), MP(5,20), MP(0,20), MP(0,0)

Parameter:
The letter to be drawn

14. The C++ random function "rand()" produces a random integer >= 0. Which of the following code segments best represents the product of the outcome of three unbiased dice each rolled independently?

a. int x = 1 + rand()%6; cout << x*x*x;
b. int x = rand()+1; y = rand()+1; cout << x*y*x;
c. int x = rand()+1; int y = x; int z = y; cout << x * y * z;
d. int x = rand()+1; int y = rand()+1; int z = rand()+1; cout << x * y * z;

Parameters:
<u>The expression</u>

15. It would be most appropriate to use a recursive function or procedure to solve a problem that

a. can be reduced to two, or more, simpler or smaller cases of the same problem
b. involves a substantial number of conditionals and nested loops
c. requires a lot of memory
d. involves storing data in a two-dimensional array
e. involves evaluation of the factorial function

Parameters:
Understanding recursion

16. The recursive sequence defined by

$x_P = 1$ if $p < 2$,
$x_n = x_{n-2} + 3$ if $n >= 2$

is best expressed by the following C++ function:

a.

```
int f ( int n )
{
        if ( n >= 2 )
                return 1;
        else
                return f( n-1 ) + 3;
}
```

b.

```
int f ( int n )
{
        if ( n < 2 )
                return f( n-2 ) + 3;
        else
                return 1;
}
```

c.

```
int f ( int n )
{
        if ( n < 2 )
                return 1;
        else
                return f( n-1 ) + 3;
}
```

d.

```
int f ( int n )
{
        if ( n < 2 )
                return 1;
        else
                return f( n-2 ) + 3;
}
```

Part Three
DATA STRUCURES
Study guide

Coverage:

1. Stacks, queues, bags, sets, rings, decques, and other abstract data types for structures. You will need to know the operations for stacks and queues. Other abstract data types will be fully explained.
2. Binary trees, binary search trees, tree balancing, B-trees, and heaps.
3. Linked lists
4. Graphs and their algorithms.
5. Object-oriented programming.
6. Sorting and Searching including hash tables.

There is very little code in this section of the comprehensive; however, where there is, it is in C++. There may be questions about why one would choose one algorithm over another or why one would choose a data structure over another. The number of items listed below does not necessarily correspond to the exact assignment of questions.

1. *Let Q be an empty queue (first-in-first-out) and S an empty stack. If elements A1, A2, A3 are placed in the queue and elements A4 and A5 are pushed onto the stack, and if all the elements from the queue are then dequeued and placed onto the stack, which element is on the top of the stack?*

   Parameters:
   This question can be varied quite widely. One could move other elements between the stack and the queue and ask what is at the rear or front of the queue and what is on the top of the stack. The question can be varied to describe algorithms to reverse the elements of a stack using only other stacks, reverse a queue, or to split data from a stack into queues or merge queue values into a stack (or vice versa).

2. *Given an array of integer values, determine if it is a heap (min heap or max heap).*

   Parameters:
   Given a heap, delete the top element and reform it as a heap, or add an element to the rear of the heap and reform it into a heap – using the usual algorithms. Describe a heap sort.

3. *Comparing Data Structures: Explain why using a heap for a priority queue is preferable, in general, to using a linked list where elements are inserted in order.*

   Parameters:
   Other comparison of data structures can be made: compare the implementations of a linked list (double, simple, circular, use of a head node, etc.); compare using an array for a queue with using a linked list for a queue (or stack); compare using an adjacency list for a graph with using an adjacency list; queues can be implemented by a simple (ending in NULL) single linked list with two pointers (to the head and tail) and as a single pointer to the rear, where the next pointer in the read points to the head (why is the second a better choice?); etc.

4. Tree Balancing: Given a binary search tree, draw a balanced version of the same tree (as a perfect binary search tree).

   Parameters:
   Given a list of integers (no more than 10) and an empty binary search tree, create an AVL tree from that list.

5. *Binary Tree Traversals: Given a binary tree with integer nodes, if traversing is writing out the node values, describe a preorder (or postorder or inorder) traversal.*

6. *Operations on Binary Search Trees: Given a list of integers and an empty binary search tree, create a binary search tree containing these integers (in the order they are listed).*

   Parameters:
   Given a tree show the result of deleting an element from the tree. Describe an algorithm for deleting all the elements greater than a specific real number from the tree. Describe an algorithm for counting all the elements in the tree that have a specific quality (like greater than 50, odd, etc.). Describe an algorithm for finding the height (or largest element, average, etc.).

7. *Given an adjacency matrix for a graph, find the transitive closure adjacency matrix.*

   Parameters:
   Given an adjacency matrix for a graph (or a picture of the graph), find a minimal spanning tree (matrix or picture); find a topological sort, find the shortest path between two specific nodes, find the precedence array that describes all shortest paths from a specific node.

8. Given a class and code using the class:

```
class Decque {  //pronounced “DECK”
public
  Decque(); // Constructor
  void AddToFront( int item );
  void AddToRear( int item );
  int RemoveFront();  //remove and
    //returns front element
  int RemoveRear();  //remove and
    //returns read element
public:
…
};
```

```
Decque d;
for ( int j = 1; j <= 10; j++ )
  if ( j % 3 == 0 )
    d.AddToFront( j );
  else
    d.AddToRear( j );
int GET;
for ( j = 1; j <= 5; j++ )
{
  GET = d.RemoveRear();
  if ( GET % 2 == 0 )
    d.AddToFront( GET );
}
```

How many elements are on the decque? What element is at the rear of the decque? What element is at the front of the deque?

Parameters:
The class can be a stack, queue, decque, bag, or some other structure. The question will be what is in the structure after the code is executed (or what is at a certain position (front, rear, top, etc.), how many elements are in the structure, etc.). When are constructors and destructors executed?

9. Comparison and Evaluation of Algorithms: Why would an insertion sort be faster than a quick sort?

Parameters:
What is the effect of using a recursive algorithms verses the iterative counterpart? How can sorting algorithms be effected by certain parameters (pivot choices in quick sort, size of data elements, etc.). What is the effect on search speed to the size of a hash table, the data key being stored, the “quality” of the hash function, the collision resolution algorithm, etc.? Why would a recursive version of an algorithm be better than the iterative version? The interpolation search has fewer comparisons in general than a binary search; under what 2 conditions might a binary search be faster.

Part Four
Computer Theories
Study guide

Coverage:

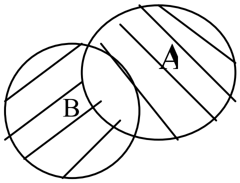
The material in this section is covered in CSC270 Discrete Structures and CSC361 Survey of Programming Languages. Texts for these courses are good references to use.

Sets
Logic
    Propositional Calculus; predicate calculus; conjunctive normal form;
    implications; converse; contrapositive; inverse; negation of an implication
Proof methods, proof by induction
Functions
Relations
    Reflexive, symmetric, non-symmetric, transitive
    Equivalence relations
    Order relations, total order, partial order, topological ordering, hasse diagrams
    Database relations: selection, projection, join
Trees
    Binary trees, preorder, inorder, post order
Graphs digraphs, paths
Permutations, combinations

Lexical definition, syntax, and semantics of a programming language
Regular expressions
Grammars
Interpreted languages, compiled languages
Families of programming languages; Procedural, functional, distributed, logic, object oriented
Expressions and operators, operator precedence, associative rules
Lisp, Prolog, simple programs and recursion
Block structure
Run time stack, activation records, static chain
Parameters, formal, actual
Parameter passing, by name, by value, by reference, by copy in/copy out

Sample Problems

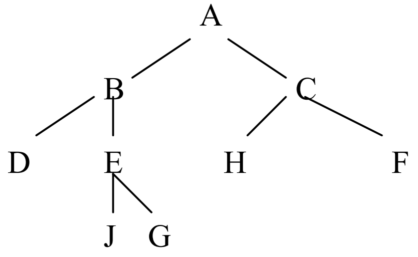
1. The set expression of the shaded area of the figure on right is



a. A union B minus A intersect B
b. A union B minus A union B
c. A minus B
d. None of the above

Parameter: The Figure given

2. If the binary tree shown below is traversed in post order, in what order are the nodes visited?



a. ABDEJCFHG
b. DJGEBHFAC
c. DJGEBHFCA
d. ABCDEFGHI
e. JHGDEFBCA

Parameter: The tree and the traverse order

3. The relation defined on {(0, 1, 2)} by {(0,0), (1,1), (0,1), (1, 2)}

1. Reflexive
2. Symmetric
3. Transitive

a. 1 only

b. both 1 and 2
c. all of them
d. 2 only
e. 3 only

Parameter; The relation definition

4. A language L is defined in the alphabet {(0,1) by L = {010,0110, 01110,….}. Which regular expression generates L?

a. (01*1)
b. (011+0)
c. (0*1*0*)
d. (011*0)
e. none of the above

Parameter: The language L

5. A committee of 3 faculty members and two students can be selected from 7 faculty members and 8 students in how many different ways?

a. 490
b. 1260
c. 980
d. 540
e. none of the above

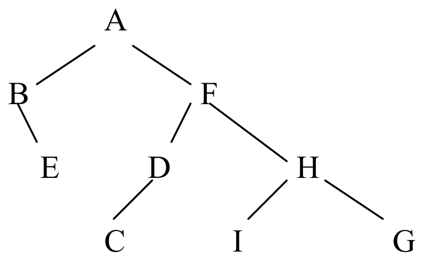
Parameter: The committee set-up

6. What is the postfix expression of 3 * [(m – n) + 5 * (s – t)]?

a. **3 m n – 5 s t - * + ***
b. * 3 – m n + * 5 – s t
c. m – n + s – t * 5 * 3
d. * 3 + m n - * 5 – s t

Parameter: The expression

7. Which of the following is the preorder of the given tree



a. DBEFAGCHI
b. ABDEFCGHI
c. ABEFDCHIG
d. DBAFEGCIH

Parameter: The tree, and the order

8. Which language is designed specially for embedded computer system?

a. FORTRAN
b. Ada
c. SNOBOL
d. LISP
e. C

Parameter: the designed specialty and the language.

9. Which of the following parameter passing mechanisms is being used to retrieve the value of "result" from function Calc?

```
void Calc( int a, int b, int & c );
int main()
{
        int s = 9; t = 99; result;
        Calc( s, t, result );
        return 1;
}
void Calc( int a, int b, int & c )
{
        c = a + b;
}
```

a. Call by vlaue
b. Call by reference
c. Call by name
d. Call by value-result
e. Call by number

Parameter: The program and the mechanism purpose.

10. A grammar is described as follow:

S → aS
S → b
S → bA
A →bB
B → a

Which of the following strings can not be derived from the above grammar?

a. ab
b. abba
c. abbb
d. bba
e. aab

Parameter: The grammar

11. What would the output of the following program be, if it is executed using dynamic scoping?

```
PROGRAM S;
VAR X : INTEGER;
PROCEDURE P;
BEGIN
                    X : = 3;
                    WRITE (X);
END;
PROCEDURE Q;
VAR X : INTEGER;
BEGIN
          X := 15;
          P;
          WRITE (X);
END;

BEGIN
          X := 10;
          Q;
          WRITE (X);
END.
```

a. 10   15   3
b**. 3   3   10**
c. 3   15   10
d. 3   3   15

Parameter: The program and the scoping