Comprehensive Study Notes and Practice Questions

BCA Data Analyst Role Examination Preparation

Day 1: Programming Fundamentals & Data Structures Foundation

C Programming Essentials - Study Notes

1. Data Types and Variables

- **Primitive Data Types:**
  - `int` (2-4 bytes): Integer values
  - `float` (4 bytes): Single precision decimal
  - `double` (8 bytes): Double precision decimal
  - `char` (1 byte): Single character
- **Type Modifiers:**
  - `signed`, `unsigned`
  - `short`, `long`
  - `const`, `volatile`
- **Variable Declaration and Initialization:**

```
int age = 25;
float salary = 50000.75;
char grade = 'A';
const int MAX_SIZE = 100;
```

2. Operators and Expressions

- **Arithmetic Operators:** `+`, `-`, `*`, `/`, `%`
- **Relational Operators:** `<`, `>`, `<=`, `>=`, `==`, `!=`
- **Logical Operators:** `&&`, `||`, `!`
- **Assignment Operators:** `=`, `+=`, `-=`, `*=`, `/=`
- **Increment/Decrement:** `++`, `--`

Operator Precedence (High to Low):

1. `()` - Parentheses

2. `++`, `--` - Postfix
3. `++`, `--`, `!` - Prefix
4. `*`, `/`, `%`
5. `+`, `-`
6. `<`, `<=`, `>`, `>=`
7. `==`, `!=`
8. `&&`
9. `||`
10. `=`, `+=`, etc.

3. Control Structures

- **Selection:** `if-else`, `switch-case`
- **Iteration:** `while`, `do-while`, `for`
- **Jump:** `break`, `continue`, `goto`, `return`

4. Arrays and Strings

- **Array Declaration:**

```
int arr[10];           // 1D array
int matrix[3][4];      // 2D array
int arr[] = {1,2,3,4}; // Initialization
```

- **String Operations:**

```
char str[20] = "Hello";
strlen(str);    // Length
strcpy(dest, src);  // Copy
strcat(dest, src);  // Concatenate
strcmp(str1, str2); // Compare
```

5. Pointers

- **Declaration:** `int *ptr;`
- **Initialization:** `ptr = &variable;`
- **Dereferencing:** `*ptr`
- **Pointer Arithmetic:** `ptr++`, `ptr--`, `ptr+n`

Key Concepts:

- Call by Value vs Call by Reference
- Dynamic Memory Allocation: `malloc()`, `calloc()`, `realloc()`, `free()`
- Pointer to Array vs Array of Pointers

Data Structures - Study Notes

1. Arrays

- **Operations:** Insert, Delete, Search, Traverse
- **Time Complexity:**
  - Access: O(1)
  - Search: O(n)
  - Insertion: O(n)
  - Deletion: O(n)

2. Linked Lists

- **Types:** Singly, Doubly, Circular
- **Operations:** Insert, Delete, Search, Reverse
- **Advantages:** Dynamic size, efficient insertion/deletion
- **Disadvantages:** Extra memory for pointers, no random access

3. Stacks

- **LIFO:** Last In, First Out
- **Operations:** Push, Pop, Peek/Top, isEmpty
- **Applications:** Expression evaluation, recursion, backtracking
- **Time Complexity:** All operations O(1)

4. Queues

- **FIFO:** First In, First Out
- **Types:** Simple, Circular, Priority, Double-ended
- **Operations:** Enqueue, Dequeue, Front, Rear
- **Applications:** BFS, scheduling, buffering

Practice Questions - Day 1

C Programming MCQs

1. What is the size of `int` data type in most 32-bit systems?
   a) 2 bytes b) 4 bytes c) 8 bytes d) 1 byte
2. Which operator has the highest precedence?
   a) `*` b) `+` c) `()` d) `=`

3. What will be the output of: `printf("%d", 5 + 3 * 2);`?
   a) 16 b) 11 c) 13 d) 10
4. In `int arr[5] = {1, 2};`, what are the values of arr[3] and arr[4]?
   a) Garbage values b) 0, 0 c) 1, 2 d) Compiler error
5. What does `int *ptr = &var;` do?
   a) Creates a pointer to var
   b) Creates a variable ptr
   c) Assigns address of var to ptr
   d) Both a and c

Data Structures MCQs

6. What is the time complexity of accessing an element in an array?
   a) O(1) b) O(n) c) O(log n) d) O(n²)
7. Which data structure follows LIFO principle?
   a) Queue b) Stack c) Array d) Linked List
8. In a singly linked list, to insert at the beginning, time complexity is:
   a) O(1) b) O(n) c) O(log n) d) O(n²)
9. Which operation is NOT performed on a stack?
   a) Push b) Pop c) Peek d) Dequeue
10. Circular queue is used to overcome which limitation of simple queue?
    a) Fixed size b) Waste of memory c) Complex operations d) All of the above

Day 2: Advanced Data Structures & Algorithms

Trees - Study Notes

1. Binary Trees

- **Properties:**
  - Each node has at most 2 children
  - Left and right subtrees are also binary trees
  - Maximum nodes at level i = 2^i
  - Maximum nodes in tree of height h = 2^(h+1) - 1

2. Tree Traversals

- **Inorder:** Left → Root → Right
- **Preorder:** Root → Left → Right
- **Postorder:** Left → Right → Root
- **Level Order:** BFS traversal

3. Binary Search Tree (BST)

- **Property:** Left subtree < Root < Right subtree
- **Operations:** Insert, Delete, Search
- **Time Complexity:** Average O(log n), Worst O(n)

4. AVL Trees

- **Balanced BST:** Height difference ≤ 1
- **Rotations:** Left, Right, Left-Right, Right-Left
- **Time Complexity:** All operations O(log n)

Searching and Sorting - Study Notes

Searching Algorithms

1. **Linear Search:**
   - Time: O(n), Space: O(1)
   - Works on unsorted arrays
2. **Binary Search:**
   - Time: O(log n), Space: O(1)
   - Requires sorted array

Sorting Algorithms

1. **Bubble Sort:**
   - Time: $O(n^2)$, Space: O(1)
   - Stable, adaptive
2. **Selection Sort:**
   - Time: $O(n^2)$, Space: O(1)
   - Not stable, not adaptive
3. **Insertion Sort:**
   - Time: $O(n^2)$, Space: O(1)
   - Stable, adaptive, efficient for small arrays
4. **Merge Sort:**
   - Time: O(n log n), Space: O(n)
   - Stable, divide and conquer
5. **Quick Sort:**
   - Time: Average O(n log n), Worst $O(n^2)$, Space: O(log n)

- Not stable, in-place

Practice Questions - Day 2

Trees MCQs

11. In a binary tree with n nodes, minimum possible height is:
    a) $\log_2(n+1) - 1$ b) $\log_2(n+1)$ c) $\log_2 n$ d) n-1
12. Inorder traversal of BST gives:
    a) Sorted sequence b) Reverse sorted c) Random order d) Level order
13. For the tree traversal: Preorder = ABC, Inorder = BAC, what is Postorder?
    a) BCA b) BAC c) CBA d) CAB
14. In AVL tree, balance factor can be:
    a) -1, 0, 1 b) -2, -1, 0, 1, 2 c) Any integer d) Only positive
15. Time complexity of searching in BST (average case):
    a) O(1) b) O(log n) c) O(n) d) $O(n^2)$

Searching and Sorting MCQs

16. Binary search can be applied on:
    a) Any array b) Sorted array only c) Linked list d) Stack
17. Which sorting algorithm has best case time complexity O(n)?
    a) Bubble sort b) Quick sort c) Merge sort d) Selection sort
18. Most efficient sorting algorithm for large datasets:
    a) Bubble sort b) Insertion sort c) Merge sort d) Selection sort
19. Which is NOT a stable sorting algorithm?
    a) Bubble sort b) Insertion sort c) Merge sort d) Quick sort
20. Time complexity of linear search in worst case:
    a) O(1) b) O(log n) c) O(n) d) $O(n^2)$

Day 3: Database Management Systems

Database Fundamentals - Study Notes

1. DBMS Concepts

- **Database:** Collection of interrelated data
- **DBMS:** Software to manage databases
- **Data Independence:** Logical and Physical
- **Schema:** Structure of database (External, Conceptual, Internal)

2. ER Model

- **Entity:** Real-world object (Strong/Weak)
- **Attribute:** Property of entity (Simple/Composite, Single/Multi-valued)
- **Relationship:** Association between entities
- **Cardinality:** One-to-One, One-to-Many, Many-to-Many

3. Relational Model

- **Relation:** Table with rows and columns
- **Tuple:** Row in a table
- **Attribute:** Column in a table
- **Domain:** Set of allowed values
- **Keys:** Super, Candidate, Primary, Foreign, Alternate

SQL - Study Notes

1. Data Definition Language (DDL)

```
CREATE TABLE Students (
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    age INT CHECK (age > 0)
);

ALTER TABLE Students ADD COLUMN email VARCHAR(100);
DROP TABLE Students;
```

2. Data Manipulation Language (DML)

```
INSERT INTO Students VALUES (1, 'John', 20, 'john@email.com');
UPDATE Students SET age = 21 WHERE id = 1;
DELETE FROM Students WHERE age < 18;
```

3. Data Query Language (DQL)

```
SELECT name, age FROM Students WHERE age > 18;
SELECT * FROM Students ORDER BY name ASC;
SELECT COUNT(*) FROM Students GROUP BY age HAVING COUNT(*) > 1;
```

4. Joins

- **INNER JOIN:** Matching rows from both tables
- **LEFT JOIN:** All rows from left table
- **RIGHT JOIN:** All rows from right table
- **FULL OUTER JOIN:** All rows from both tables

Normalization - Study Notes

Normal Forms

1. **1NF:** Atomic values, no repeating groups
2. **2NF:** 1NF + No partial dependencies
3. **3NF:** 2NF + No transitive dependencies
4. **BCNF:** 3NF + Every determinant is a candidate key

Practice Questions - Day 3

Database Concepts MCQs

21. Which is NOT a characteristic of DBMS?
    a) Data redundancy b) Data independence c) Data security d) Data integrity
22. In ER model, a relationship between entity and itself is called:
    a) Unary b) Binary c) Ternary d) Recursive
23. Which key uniquely identifies a tuple in a relation?
    a) Foreign key b) Super key c) Primary key d) Candidate key
24. ACID properties include all EXCEPT:
    a) Atomicity b) Consistency c) Isolation d) Availability
25. Cardinality of a relation refers to:
    a) Number of attributes b) Number of tuples c) Number of tables d) Number of keys

SQL MCQs

26. Which SQL command is used to remove a table?
    a) DELETE b) REMOVE c) DROP d) CLEAR
27. To retrieve unique values, which keyword is used?
    a) UNIQUE b) DISTINCT c) DIFFERENT d) SEPARATE
28. GROUP BY clause is used with:
    a) Aggregate functions b) WHERE clause c) ORDER BY d) All of the above
29. Which join returns all rows from both tables?
    a) INNER JOIN b) LEFT JOIN c) RIGHT JOIN d) FULL OUTER JOIN

30. To add a new column to existing table:
    a) ADD COLUMN b) ALTER TABLE...ADD c) INSERT COLUMN d) MODIFY TABLE

Normalization MCQs

31. A table is in 1NF if:
    a) It has a primary key
    b) It has atomic values
    c) It has no redundancy
    d) It has foreign keys
32. 2NF eliminates:
    a) Partial dependencies b) Transitive dependencies c) Both d) Neither
33. BCNF is stricter than:
    a) 1NF b) 2NF c) 3NF d) All of the above
34. Functional dependency A → B means:
    a) B determines A b) A determines B c) A equals B d) A and B are independent
35. Normalization helps in:
    a) Reducing redundancy b) Avoiding anomalies c) Saving space d) All of the above

Day 4: Operating Systems

OS Fundamentals - Study Notes

1. Operating System Concepts

- **Definition:** Interface between user and hardware
- **Functions:** Process management, memory management, file management, I/O management
- **Types:** Batch, Time-sharing, Real-time, Distributed, Mobile

2. Process Management

- **Process:** Program in execution
- **Process States:** New, Ready, Running, Waiting, Terminated
- **Process Control Block (PCB):** Process state, program counter, registers, memory limits

CPU Scheduling - Study Notes

Scheduling Algorithms

1. **First Come First Serve (FCFS):**
    - Non-preemptive
    - Average waiting time can be high

- Simple to implement

2. **Shortest Job First (SJF):**
   - Optimal for minimum average waiting time
   - Can be preemptive (SRTF) or non-preemptive
3. **Round Robin (RR):**
   - Preemptive
   - Uses time quantum
   - Fair scheduling
4. **Priority Scheduling:**
   - Higher priority processes executed first
   - Can cause starvation

Memory Management - Study Notes

Memory Allocation Techniques

1. **Contiguous Allocation:**
   - Fixed partitioning
   - Variable partitioning
   - Internal and external fragmentation
2. **Paging:**
   - Fixed-size pages
   - Page table for address translation
   - No external fragmentation
3. **Segmentation:**
   - Variable-size segments
   - Segment table
   - Logical division of program
4. **Virtual Memory:**
   - More programs in memory than physical capacity
   - Page replacement algorithms: FIFO, LRU, Optimal

Practice Questions - Day 4

OS Concepts MCQs

36. Which is NOT a function of operating system?
    a) Process management b) Memory management c) Code compilation d) File management
37. In which state does a process wait for I/O completion?
    a) Ready b) Running c) Waiting d) Terminated
38. Process Control Block contains:
    a) Process state b) Program counter c) CPU registers d) All of the above
39. Context switching occurs between:
    a) Processes b) Threads c) Both d) Neither
40. Kernel is:
    a) Hardware component b) Core of OS c) Application program d) Device driver

CPU Scheduling MCQs

41. FCFS scheduling may cause:
    a) Starvation b) Convoy effect c) Priority inversion d) Deadlock
42. Which algorithm gives minimum average waiting time?
    a) FCFS b) SJF c) Round Robin d) Priority
43. In Round Robin, if time quantum is very large, it becomes:
    a) FCFS b) SJF c) Priority d) Multilevel
44. Preemptive scheduling means:
    a) Process cannot be interrupted
    b) Process can be interrupted
    c) Process runs to completion
    d) Process waits indefinitely
45. Starvation can occur in:
    a) FCFS b) SJF c) Priority scheduling d) Round Robin

Memory Management MCQs

46. Paging suffers from:
    a) External fragmentation b) Internal fragmentation c) Both d) Neither
47. Page fault occurs when:
    a) Page is in memory b) Page is not in memory c) Memory is full d) Process terminates
48. LRU stands for:
    a) Least Recently Used b) Last Recently Used c) Latest Recent Update d) Logical Resource Unit
49. Virtual memory allows:
    a) Larger programs b) More programs c) Both d) Neither
50. Translation Lookaside Buffer (TLB) is used for:
    a) Address translation b) Memory allocation c) Process scheduling d) I/O operations

Answer Key for Practice Questions

Day 1 (Questions 1-10):

1. b) 4 bytes
2. c) ()
3. b) 11
4. b) 0, 0
5. d) Both a and c
6. a) O(1)
7. b) Stack
8. a) O(1)
9. d) Dequeue
10. b) Waste of memory

Day 2 (Questions 11-20):

11. a) $\log_2(n+1) - 1$
12. a) Sorted sequence
13. a) BCA
14. a) -1, 0, 1
15. b) O(log n)
16. b) Sorted array only
17. a) Bubble sort
18. c) Merge sort
19. d) Quick sort
20. c) O(n)

Day 3 (Questions 21-35):

21. a) Data redundancy
22. d) Recursive
23. c) Primary key
24. d) Availability
25. b) Number of tuples
26. c) DROP
27. b) DISTINCT
28. a) Aggregate functions
29. d) FULL OUTER JOIN
30. b) ALTER TABLE...ADD
31. b) It has atomic values
32. a) Partial dependencies
33. c) 3NF
34. b) A determines B
35. d) All of the above

Day 4 (Questions 36-50):
36. c) Code compilation
37. c) Waiting
38. d) All of the above
39. c) Both
40. b) Core of OS
41. b) Convoy effect
42. b) SJF
43. a) FCFS
44. b) Process can be interrupted
45. c) Priority scheduling
46. b) Internal fragmentation
47. b) Page is not in memory
48. a) Least Recently Used
49. c) Both
50. a) Address translation

Quick Reference Formulas

Data Structures

- Array access: O(1)
- Linear search: O(n)
- Binary search: O(log n)
- Tree height: $\log_2(n+1)$ - 1 to n-1

Database

- Functional dependency: X → Y
- BCNF condition: Every determinant is a candidate key

Operating Systems

- Turnaround time = Completion time - Arrival time
- Waiting time = Turnaround time - Burst time
- Response time = First execution time - Arrival time

Memory Management

- Page size = 2^k bytes
- Number of pages = Process size / Page size
- Internal fragmentation = Page size - Process size in last page