Complete BCA Data Analyst Examination Preparation Guide

Comprehensive Study Notes, Practice Questions & Mock Tests

Table of Contents

7-Day Study Plan

Optimal Schedule for BCA Data Analyst Role Examination

Day 1: Programming Fundamentals (8 hours)

- **Morning (3 hours):** C Programming basics, data types, operators
- **Afternoon (3 hours):** Control structures, arrays, strings
- **Evening (2 hours):** Practice MCQs (Questions 1-10)

Day 2: Data Structures & Algorithms (8 hours)

- **Morning (3 hours):** Arrays, linked lists, stacks, queues
- **Afternoon (3 hours):** Trees, searching, sorting algorithms
- **Evening (2 hours):** Practice MCQs (Questions 11-20)

Day 3: Database Management Systems (8 hours)

- **Morning (3 hours):** ER model, relational algebra, SQL
- **Afternoon (3 hours):** Normalization, indexing, transactions
- **Evening (2 hours):** Practice MCQs (Questions 21-35)

Day 4: Operating Systems (8 hours)

- **Morning (3 hours):** Process management, CPU scheduling
- **Afternoon (3 hours):** Memory management, file systems
- **Evening (2 hours):** Practice MCQs (Questions 36-50)

Day 5: Python Programming (8 hours)

- **Morning (3 hours):** Python basics, data structures, OOP
- **Afternoon (3 hours):** File handling, exception handling
- **Evening (2 hours):** Practice MCQs (Questions 51-75)

Day 6: Java Programming (8 hours)

- **Morning (3 hours):** Java fundamentals, OOP concepts
- **Afternoon (3 hours):** Collections, exception handling, multithreading
- **Evening (2 hours):** Practice MCQs (Questions 76-100)

Day 7: Revision & Mock Tests (8 hours)

- **Morning (2 hours):** Quick revision of all topics
- **Afternoon (3 hours):** Mock Test Paper 1
- **Evening (3 hours):** Mock Test Paper 2 & Final Review

Day 1-2: Programming Fundamentals & Data Structures

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

5. 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

6. Tree Traversals

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

7. Binary Search Tree (BST)

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

8. Searching and Sorting Algorithms

Searching:

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

Sorting:

1. **Bubble Sort:** Time: $O(n^2)$, Space: O(1) - Stable, adaptive
2. **Selection Sort:** Time: $O(n^2)$, Space: O(1) - Not stable
3. **Insertion Sort:** Time: $O(n^2)$, Space: O(1) - Stable, adaptive
4. **Merge Sort:** Time: O(n log n), Space: O(n) - Stable
5. **Quick Sort:** Time: Average O(n log n), Worst $O(n^2)$, Space: O(log n)

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

4. SQL - Structured Query Language

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;
```

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;
```

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;
```

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

5. Normalization

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

6. Transactions and Concurrency

- **ACID Properties:**
  - **Atomicity:** All or nothing
  - **Consistency:** Database remains in consistent state
  - **Isolation:** Transactions don't interfere
  - **Durability:** Changes persist after commit

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

3. CPU Scheduling

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

4. Memory Management

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

5. File Systems

- **File Operations:** Create, Delete, Read, Write, Seek
- **Directory Structure:** Single-level, Two-level, Tree-structured
- **File Allocation Methods:** Contiguous, Linked, Indexed

6. Deadlock

- **Necessary Conditions:** Mutual exclusion, Hold and wait, No preemption, Circular wait
- **Prevention:** Negate one of the necessary conditions
- **Avoidance:** Banker's algorithm
- **Detection and Recovery:** Resource allocation graph

Day 5: Python Programming

Python Fundamentals - Study Notes

1. Python Basics

- **Features:** Interpreted, Object-oriented, High-level, Dynamic typing
- **Python Execution:** Source code → Bytecode → Python Virtual Machine
- **Memory Management:** Automatic garbage collection, reference counting

2. Data Types

```
# Primitive Types
integer = 42
floating = 3.14
string = "Hello World"
boolean = True

# Collections
list_data = [1, 2, 3, 4]
tuple_data = (1, 2, 3, 4)
dict_data = {"key": "value", "age": 25}
set_data = {1, 2, 3, 4}
```

3. Control Structures

```
# Conditional
if condition:
    statement
elif another_condition:
    statement
else:
    statement

# Loops
for item in iterable:
    statement

while condition:
    statement

# List Comprehension
squares = [x**2 for x in range(10)]
```

4. Functions

```
# Function Definition
def function_name(param1, param2="default"):
    """Docstring"""
    return result

# Lambda Functions
square = lambda x: x**2
filter(lambda x: x > 0, numbers)
map(lambda x: x*2, numbers)
```

5. Object-Oriented Programming

```
class ClassName:
    class_variable = "shared"

    def __init__(self, param):
        self.instance_variable = param

    def method(self):
        return self.instance_variable

    @staticmethod
    def static_method():
        return "static"

    @classmethod
    def class_method(cls):
        return cls.class_variable

# Inheritance
class ChildClass(ClassName):
    def __init__(self, param, extra):
        super().__init__(param)
        self.extra = extra
```

6. File Handling

```
# Reading Files
with open("file.txt", "r") as file:
    content = file.read()
    lines = file.readlines()
    line = file.readline()

# Writing Files
with open("file.txt", "w") as file:
    file.write("Hello World")
    file.writelines(["line1\n", "line2\n"])
```

7. Exception Handling

```
try:
    risky_operation()
except SpecificError as e:
    handle_specific_error(e)
except Exception as e:
    handle_general_error(e)
else:
    runs_if_no_exception()
finally:
    always_runs()
```

Day 6: Java Programming

Java Fundamentals - Study Notes

1. Java Basics

- **Features:** Platform independent, Object-oriented, Strongly typed
- **JVM Architecture:** Class Loader, Runtime Data Area, Execution Engine
- **Compilation:** Source code (.java) → Bytecode (.class) → JVM execution

2. Data Types and Variables

```
// Primitive Types
int number = 42;
double decimal = 3.14;
char character = 'A';
boolean flag = true;

// Reference Types
String text = "Hello World";
int[] array = {1, 2, 3, 4};
```

3. Object-Oriented Programming

```
// Class Definition
public class Person {
    private String name;
    private int age;

    // Constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
```

```
    // Getter/Setter
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    // Method
    public void displayInfo() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

// Inheritance
public class Student extends Person {
    private String studentId;

    public Student(String name, int age, String studentId) {
        super(name, age); // Call parent constructor
        this.studentId = studentId;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Student ID: " + studentId);
    }
}
```

4. Interfaces and Abstract Classes

```
// Interface
public interface Drawable {
    void draw(); // Abstract method

    default void display() { // Default method (Java 8+)
        System.out.println("Displaying...");
    }
}

// Abstract Class
public abstract class Shape {
    protected String color;

    public abstract double area(); // Abstract method

    public void setColor(String color) { // Concrete method
        this.color = color;
    }
}
```

5. Exception Handling

```
try {
    riskyOperation();
} catch (SpecificException e) {
    System.out.println("Specific error: " + e.getMessage());
} catch (Exception e) {
    System.out.println("General error: " + e.getMessage());
} finally {
    cleanupCode();
}

// Custom Exception
public class CustomException extends Exception {
    public CustomException(String message) {
        super(message);
    }
}
```

6. Collections Framework

```
// List
List<String> list = new ArrayList<>();
list.add("Item");
list.get(0);
list.size();

// Set
Set<String> set = new HashSet<>();
set.add("Unique");
set.contains("Unique");

// Map
Map<String, Integer> map = new HashMap<>();
map.put("key", 100);
map.get("key");
```

Mock Test Paper 1: Comprehensive Assessment

Time: 2 Hours | Total Questions: 100 | Marks: 200

Section A: Programming Fundamentals (25 Questions)

1. Which of the following is NOT a valid C data type?
   a) signed int b) unsigned float c) long double d) short char
2. What is the output of the following C code?

```
int x = 5;
printf("%d", x++);
```

a) 5 b) 6 c) Compiler error d) Undefined behavior

3. In Python, which of the following is immutable?
   a) List b) Dictionary c) Set d) Tuple
4. Java method overriding requires:
   a) Same method name only
   b) Same method name and parameters
   c) Same return type only
   d) Inheritance relationship
5. Which operator has the highest precedence in C?
   a) * b) + c) () d) &&
6. Python `__init__` method is called:
   a) When class is defined b) When object is created c) When method is called d) When object is deleted
7. In Java, `ArrayList` is part of which package?
   a) java.lang b) java.util c) java.io d) java.net
8. What does the following Python code output?

```
print([i for i in range(3)])
```

   a) [1, 2, 3] b) [0, 1, 2] c) [0, 1, 2, 3] d) Error
9. C pointer arithmetic: If `int *p` points to address 1000, what is `p + 2`?
   a) 1002 b) 1004 c) 1008 d) Depends on system
10. Java `static` methods can access:
    a) Instance variables only b) Static variables only c) Both d) Neither
11. Time complexity of inserting an element at the beginning of a linked list:
    a) O(1) b) O(n) c) O(log n) d) $O(n^2)$
12. Which traversal of BST gives sorted output?
    a) Preorder b) Inorder c) Postorder d) Level order
13. In a max heap with 15 elements, what is the maximum number of comparisons needed to find the minimum element?
    a) 1 b) 7 c) 8 d) 14
14. Which sorting algorithm is most efficient for already sorted arrays?
    a) Bubble sort b) Quick sort c) Merge sort d) Insertion sort
15. Hash table with separate chaining handles collisions by:
    a) Linear probing b) Quadratic probing c) Linked lists d) Double hashing
16. In a binary tree, maximum number of nodes at level 4 is:
    a) 8 b) 15 c) 16 d) 31
17. Stack is best suited for:
    a) BFS traversal b) Expression evaluation c) Finding shortest path d) Sorting
18. Time complexity of deleting an element from the middle of an array:
    a) O(1) b) O(log n) c) O(n) d) $O(n^2)$

19. Which of the following uses FIFO principle?
    a) Stack b) Queue c) Tree d) Graph
20. Binary search requires:
    a) Sorted array b) Complete binary tree c) Hash table d) Linked list

Section B: Database Management (25 Questions)

21. In ER model, a weak entity:
    a) Has its own primary key b) Depends on strong entity c) Cannot have attributes d) All of the above
22. SQL command to remove all records from a table without deleting the table structure:
    a) DROP b) DELETE c) TRUNCATE d) REMOVE
23. Which normal form eliminates transitive dependencies?
    a) 1NF b) 2NF c) 3NF d) BCNF
24. INNER JOIN returns:
    a) All records from left table
    b) All records from right table
    c) Only matching records from both tables
    d) All records from both tables
25. In ACID properties, 'C' stands for:
    a) Concurrency b) Consistency c) Completeness d) Correctness
26. Primary key constraint ensures:
    a) Uniqueness only b) Not null only c) Both uniqueness and not null d) Foreign key reference
27. Which SQL clause is used to sort result set?
    a) GROUP BY b) HAVING c) ORDER BY d) WHERE
28. Index in database is used for:
    a) Data storage b) Fast retrieval c) Data validation d) Transaction control
29. Deadlock in database occurs when:
    a) Transaction takes too long
    b) Two transactions wait for each other
    c) Database runs out of memory
    d) Index is corrupted
30. VIEW in SQL is:
    a) Physical table b) Virtual table c) Index d) Constraint
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

Section C: Operating Systems (25 Questions)

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
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
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

Section D: Python and Java (25 Questions)

51. Python is:
    a) Compiled language b) Interpreted language c) Both d) Neither
52. Which is mutable in Python?
    a) Tuple b) String c) List d) Integer
53. What is the output of `print(type([]))`?
    a) `<class 'array'>` b) `<class 'list'>` c) `<class 'tuple'>` d) `<class 'dict'>`
54. List comprehension `[x for x in range(5)]` produces:
    a) [1,2,3,4,5] b) [0,1,2,3,4] c) [0,1,2,3,4,5] d) Error
55. Which method adds element to end of list?
    a) add() b) append() c) insert() d) extend()
56. Java is:
    a) Platform dependent b) Platform independent c) Hardware specific d) OS specific
57. Which is NOT a Java primitive type?
    a) int b) double c) String d) boolean
58. `public static void main(String[] args)` - 'static' means:
    a) Method belongs to class b) Method belongs to object c) Method is final d) Method is private
59. Java constructor:
    a) Has return type b) Has no return type c) Returns object d) Returns void
60. Method overriding requires:
    a) Same class b) Inheritance c) Same package d) Static methods
61. `final` keyword prevents:
    a) Inheritance b) Method overriding c) Variable modification d) All of the above
62. Interface in Java:
    a) Can have constructors b) Can have instance variables c) Can have abstract methods d) Can be instantiated
63. Exception handling uses:
    a) try-catch b) try-finally c) try-catch-finally d) All of the above
64. `ArrayList` implements:
    a) List interface b) Set interface c) Map interface d) Queue interface
65. `==` operator in Java compares:
    a) Values only b) References c) Both d) Neither
66. Python `super()` function:
    a) Creates superclass b) Calls parent method c) Checks inheritance d) Creates object
67. Lambda function `lambda x, y: x + y` is equivalent to:

```
a) def func(x, y): return x + y
b) def func(x, y): x + y
c) def func(): return x + y
d) None of the above
```

68. `with` statement in Python is used for:
    a) Loops b) Conditions c) Resource management d) Functions
69. Java `synchronized` keyword:
    a) Prevents thread interference b) Allows concurrent access c) Improves performance d) None of the above
70. Python `is` operator checks:
    a) Value equality b) Type equality c) Identity d) Membership
71. Java generics provide:
    a) Type safety b) Code reusability c) Performance improvement d) All of the above
72. `finally` block in Java:
    a) Always executes b) Executes only if exception occurs c) Executes only if no exception d) May not execute
73. Python `yield` keyword is used in:
    a) Functions b) Generators c) Classes d) Modules
74. Java `clone()` method:
    a) Creates shallow copy b) Creates deep copy c) Depends on implementation d) Cannot be used
75. Which collection allows duplicate values in Java?
    a) Set b) Map c) List d) Both b and c

Mock Test Paper 2: Advanced Topics

Time: 2 Hours | Total Questions: 50 | Marks: 100

Section A: Advanced Programming (15 Questions)

76. Python decorator is used for:
    a) Modifying function behavior b) Creating classes c) Exception handling d) File operations
77. C `malloc()` returns:
    a) Size of allocated memory b) Pointer to allocated memory c) Status of allocation d) Nothing
78. Java `equals()` method should be:
    a) Reflexive b) Symmetric c) Transitive d) All of the above
79. In C, array name represents:
    a) First element b) Array size c) Base address d) Array type
80. Java interface can have:
    a) Variables only b) Methods only c) Both variables and methods d) Neither

81. Python `enumerate()` function returns:
    a) Only indices b) Only values c) Index-value pairs d) Length
82. Java `instanceof` operator:
    a) Checks object type b) Creates instance c) Compares values d) Checks inheritance
83. Python `pass` statement:
    a) Terminates program b) Does nothing c) Passes parameters d) Causes error
84. Java `String` class is:
    a) Mutable b) Immutable c) Both d) Neither
85. Python `break` statement:
    a) Exits function b) Exits loop c) Exits program d) Skips iteration
86. Which is correct way to create empty dictionary in Python?
    a) {} b) dict() c) Both a and b d) []
87. Java access modifier `protected` allows access from:
    a) Same class only b) Same package and subclasses c) Everywhere d) Same package only
88. Python `"hello".upper()` returns:
    a) "HELLO" b) "Hello" c) "hello" d) Error
89. Java static variables are:
    a) Instance specific b) Class specific c) Method specific d) Block specific
90. Python `__str__` method is used for:
    a) String representation b) String comparison c) String length d) String conversion

Section B: System Design & Networking (20 Questions)

91. OSI model has how many layers?
    a) 5 b) 6 c) 7 d) 8
92. TCP is:
    a) Connection-oriented b) Connectionless c) Both d) Neither
93. IP address 192.168.1.1 is:
    a) Public b) Private c) Multicast d) Broadcast
94. HTTP uses which port by default?
    a) 21 b) 80 c) 443 d) 25
95. DNS stands for:
    a) Domain Name System b) Data Network Service c) Digital Name Server d) Dynamic Network System
96. DHCP is used for:
    a) Domain resolution b) IP address assignment c) File transfer d) Email
97. Subnet mask 255.255.255.0 represents:
    a) /24 b) /16 c) /8 d) /32
98. TCP three-way handshake involves:
    a) SYN, ACK, FIN b) SYN, SYN-ACK, ACK c) SYN, ACK, RST d) SYN, FIN, ACK

99. Which protocol is connectionless?
    a) TCP b) UDP c) HTTP d) FTP
100. MAC address is:
    a) 32-bit b) 48-bit c) 64-bit d) 128-bit
101. OSI model layer responsible for routing:
    a) Physical b) Data Link c) Network d) Transport
102. HTTP status code 404 means:
    a) Server error b) Not found c) Unauthorized d) Forbidden
103. Subnet mask 255.255.240.0 represents:
    a) /20 b) /24 c) /16 d) /28
104. DNS uses which protocol?
    a) TCP only b) UDP only c) Both TCP and UDP d) Neither
105. Default port for HTTPS:
    a) 80 b) 443 c) 21 d) 25
106. ARP is used to find:
    a) IP address from MAC b) MAC address from IP c) Domain name from IP d) Port number
107. Which HTTP method is idempotent?
    a) POST b) PUT c) PATCH d) All of the above
108. JSON stands for:
    a) JavaScript Object Notation b) Java Serialized Object Notation c) JavaScript Online Notation d) Java Object Notation
109. RESTful API primarily uses which protocol?
    a) TCP b) UDP c) HTTP d) FTP
110. Big O notation describes:
    a) Best case complexity b) Worst case complexity c) Average case complexity d) All cases

Section C: Digital Electronics & Miscellaneous (15 Questions)

111. Binary representation of decimal 10:
    a) 1010 b) 1100 c) 1001 d) 1110
112. Two's complement of binary 1010:
    a) 0101 b) 0110 c) 1010 d) 1001
113. In digital electronics, NAND gate is:
    a) Universal gate b) Basic gate c) Derived gate d) None
114. Boolean algebra: A + A' = ?
    a) 0 b) 1 c) A d) A'
115. Which number system uses base 16?
    a) Binary b) Octal c) Decimal d) Hexadecimal
116. Logic gate that gives output 1 when inputs are different:
    a) AND b) OR c) XOR d) NAND

117. Flip-flop is used for:
    a) Combinational circuits b) Sequential circuits c) Both d) Neither
118. De Morgan's law: (A + B)' = ?
    a) A' + B' b) A' . B' c) A . B d) A + B
119. Multiplexer is also called:
    a) Data selector b) Data distributor c) Decoder d) Encoder
120. Agile methodology emphasizes:
    a) Documentation b) Customer collaboration c) Contract negotiation d) Following plans
121. Software testing phase that tests individual components:
    a) Unit testing b) Integration testing c) System testing d) Acceptance testing
122. MVC architecture stands for:
    a) Model View Controller b) Multiple View Control c) Main View Container d) Master View Client
123. SDLC phase that comes after design:
    a) Analysis b) Implementation c) Testing d) Maintenance
124. HTML5 semantic tag for navigation:
    a) `<nav>` b) `<navigation>` c) `<menu>` d) `<links>`
125. CSS property for background color:
    a) bg-color b) background-color c) color-background d) bgcolor

Answer Keys

Mock Test Paper 1 (Questions 1-75)

1. b) unsigned float
2. a) 5
3. d) Tuple
4. d) Inheritance relationship
5. c) ()
6. b) When object is created
7. b) java.util
8. b) [0, 1, 2]
9. c) 1008
10. b) Static variables only
11. a) O(1)
12. b) Inorder
13. d) 14
14. d) Insertion sort

15. c) Linked lists
16. c) 16
17. b) Expression evaluation
18. c) O(n)
19. b) Queue
20. a) Sorted array
21. b) Depends on strong entity
22. c) TRUNCATE
23. c) 3NF
24. c) Only matching records from both tables
25. b) Consistency
26. c) Both uniqueness and not null
27. c) ORDER BY
28. b) Fast retrieval
29. b) Two transactions wait for each other
30. b) Virtual table
31. b) It has atomic values
32. a) Partial dependencies
33. c) 3NF
34. b) A determines B
35. d) All of the above
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
51. b) Interpreted language
52. c) List
53. b) `<class 'list'>`
54. b) [0,1,2,3,4]
55. b) append()
56. b) Platform independent
57. c) String
58. a) Method belongs to class
59. b) Has no return type
60. b) Inheritance
61. d) All of the above
62. c) Can have abstract methods
63. d) All of the above
64. a) List interface
65. b) References
66. b) Calls parent method
67. a) def func(x, y): return x + y
68. c) Resource management
69. a) Prevents thread interference
70. c) Identity
71. d) All of the above
72. d) May not execute
73. b) Generators
74. c) Depends on implementation
75. c) List

Mock Test Paper 2 (Questions 76-125)

76. a) Modifying function behavior
77. b) Pointer to allocated memory
78. d) All of the above
79. c) Base address
80. c) Both variables and methods
81. c) Index-value pairs
82. a) Checks object type

83. b) Does nothing
84. b) Immutable
85. b) Exits loop
86. c) Both a and b
87. b) Same package and subclasses
88. a) "HELLO"
89. b) Class specific
90. a) String representation
91. c) 7
92. a) Connection-oriented
93. b) Private
94. b) 80
95. a) Domain Name System
96. b) IP address assignment
97. a) /24
98. b) SYN, SYN-ACK, ACK
99. b) UDP
100. b) 48-bit
101. c) Network
102. b) Not found
103. a) /20
104. c) Both TCP and UDP
105. b) 443
106. b) MAC address from IP
107. b) PUT
108. a) JavaScript Object Notation
109. c) HTTP
110. b) Worst case complexity
111. a) 1010
112. b) 0110
113. a) Universal gate
114. b) 1
115. d) Hexadecimal
116. c) XOR
117. b) Sequential circuits

118. b) A' . B'
119. a) Data selector
120. b) Customer collaboration
121. a) Unit testing
122. a) Model View Controller
123. b) Implementation
124. a) `<nav>`
125. b) background-color

Final Day Revision Checklist

Must Remember Formulas

1. **Time Complexity Rankings:** O(1) < O(log n) < O(n) < O(n log n) < O(n²) < O(2^n)
2. **Binary Tree:** Max nodes at level i = 2^i, Total nodes in tree of height h = 2^(h+1) - 1
3. **Database:** ACID properties, Normal forms (1NF→2NF→3NF→BCNF)
4. **OS Scheduling:** Turnaround = Completion - Arrival, Waiting = Turnaround - Burst
5. **Networking:** IPv4 classes, Subnet calculations, OSI layers
6. **Digital:** Binary arithmetic, Boolean laws, Gate equivalents

Key Programming Concepts

1. **C:** Pointer arithmetic, Memory management, String functions
2. **Python:** Data structures, OOP, Built-in functions
3. **Java:** OOP principles, Exception handling, Collections
4. **Data Structures:** Implementation and time complexities
5. **Database:** SQL queries, Joins, Normalization
6. **OS:** Process states, Memory management, Synchronization

Last-Minute Tips

1. **Read questions carefully** - Watch for keywords like "NOT", "EXCEPT"
2. **Eliminate obviously wrong options** first
3. **For calculation problems**, verify your arithmetic
4. **Time management** - Don't spend too much time on any single question
5. **Review marked questions** if time permits
6. **Trust your first instinct** - don't change answers unless you're certain

Common Trap Questions to Watch Out For

1. **Array indexing** - Remember arrays start from 0
2. **Operator precedence** - Parentheses have highest precedence
3. **Pass by value vs reference** - Understand the difference
4. **SQL case sensitivity** - Keywords are case-insensitive, but data might be
5. **Complexity analysis** - Best vs Average vs Worst case scenarios

Quick Reference Summary

Programming Languages

- **C:** Low-level, pointers, manual memory management
- **Python:** High-level, interpreted, dynamic typing
- **Java:** Platform-independent, object-oriented, strongly typed

Data Structures Time Complexities

- **Array:** Access O(1), Search O(n), Insert/Delete O(n)
- **Linked List:** Access O(n), Search O(n), Insert/Delete O(1)
- **Stack/Queue:** All operations O(1)
- **Binary Search Tree:** Average O(log n), Worst O(n)
- **Hash Table:** Average O(1), Worst O(n)

Database Key Concepts

- **Keys:** Primary (unique + not null), Foreign (references primary key)
- **Joins:** INNER (matching records), LEFT/RIGHT (all from one side), FULL (all records)
- **Normal Forms:** 1NF (atomic), 2NF (no partial dependency), 3NF (no transitive dependency)

Operating System Essentials

- **Process States:** New → Ready → Running → Waiting/Terminated
- **Scheduling:** FCFS (simple), SJF (optimal), RR (fair), Priority (starvation risk)
- **Memory:** Paging (fixed size), Segmentation (variable size), Virtual memory

Networking Fundamentals

- **OSI Layers:** Physical, Data Link, Network, Transport, Session, Presentation, Application
- **TCP/IP:** Connection-oriented, reliable, 3-way handshake
- **UDP:** Connectionless, faster, no guarantee
- **IP Classes:** A (1-126), B (128-191), C (192-223)

Good luck with your examination! Focus on understanding concepts and applying logical reasoning to solve problems efficiently.