

## General Questions

### ◆ □PROGRAMMING FUNDAMENTALS – QUESTIONS & ANSWERS

Q1. What happens in memory when a program runs?

Answer:

When a program runs, memory is divided mainly into stack and heap. Stack stores function calls and local variables. Heap stores dynamically allocated data. OS allocates memory and manages execution.

---

Q2. Stack vs Heap memory?

Answer:

Stack is fast, automatic, and used for function calls. Heap is slower, manual/dynamic, and used for objects and dynamic memory allocation.

---

Q3. What is time complexity? Why important?

Answer:

Time complexity measures how execution time increases with input size. It helps choose efficient algorithms for large data.

---

Q4. What happens if memory is not released?

Answer:

It causes memory leak, leading to performance issues and program crash over time.

---

Q5. Compile-time vs Runtime error?

Answer:

Compile-time errors occur during compilation (syntax). Runtime errors occur during execution (division by zero, null reference).

## General Questions

---

Q6. How do you debug logical errors?

Answer:

By checking logic step-by-step, printing values, dry run, and testing with different inputs.

---

### ❖ C LANGUAGE – CORPORATE CRITICAL QUESTIONS & ANSWERS

Q1. Array vs Pointer?

Answer:

Array stores multiple values in contiguous memory. Pointer stores address of another variable.

---



Q2. Out-of-bounds array access?

Answer:

It leads to undefined behavior and may crash the program.

---

Q3. Why C is faster?

Answer:

C has no runtime overhead, direct memory access, and compiled to machine code.

---

Q4. What is malloc()?

Answer:

## General Questions

malloc() dynamically allocates memory at runtime from heap.

---

Q5. Dangling pointer?

Answer:

A pointer pointing to memory that has been freed.

---

Q6. Function call internal working?

Answer:

Parameters and return address are pushed into stack, function executes, then stack clears.

---

Q7. Call by value vs reference?



Answer:

Value passes copy. Reference passes address, allowing modification.

---

Q8. Not freeing memory?

Answer:

Memory leak occurs.

---



C – CRITICAL INTERVIEW POINTS (Must Know)

Pointer basics

Dynamic memory

## General Questions

Stack vs Heap

Arrays vs pointers

Memory leaks

---

### ◆ ☐JAVA – CORPORATE LEVEL QUESTIONS & ANSWERS

Q1. Java memory management?

Answer:

Java uses stack for method calls and heap for objects. Garbage collector manages memory.

---

Q2. JVM role?



*LTIMindtree*

Answer:

JVM executes bytecode, manages memory, and provides platform independence.

---

Q3. Array vs ArrayList?

Answer:

Array is fixed size. ArrayList is dynamic.

---

Q4. Why OOP?

Answer:

OOP improves code reusability, maintainability, and scalability.

## General Questions

---

Q5. Inheritance example?

Answer:

Vehicle → Car inherits properties.

---

Q6. Garbage collection?

Answer:

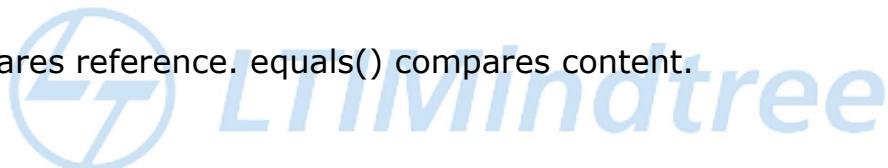
Automatically removes unused objects from memory.

---

Q7. == vs equals()?

Answer:

== compares reference. equals() compares content.



---

Q8. Constructor not written?

Answer:

Java provides default constructor.

---

### ● JAVA – IMPORTANT INTERVIEW POINTS

OOP concepts with examples

JVM + memory

Collections basics

Object vs class

## General Questions

---

### ◆ 4 PYTHON – CORPORATE LEVEL QUESTIONS & ANSWERS

Q1. Why Python slower?

Answer:

It is interpreted and dynamically typed.

---

Q2. Python execution?

Answer:

Source → bytecode → Python virtual machine.

---

Q3. List vs Tuple?



Answer:

List mutable. Tuple immutable.

---

Q4. Indentation error?

Answer:

Program fails with syntax error.

---

Q5. Compiled or interpreted?

Answer:

Both. Compiled to bytecode, then interpreted.

## General Questions

---

Q6. Dynamic typing?

Answer:

Variable type decided at runtime.

---

Q7. Python memory handling?

Answer:

Automatic memory management using reference counting.

---

Q8. Python in data science?

Answer:

Rich libraries and easy syntax.



---

### ● PYTHON – IMPORTANT INTERVIEW POINTS

Indentation

List, tuple, dict

Dynamic typing

Execution model

---

### ◆ SQL – CORPORATE / REAL USAGE ANSWERS

Q1. WHERE vs HAVING?

## General Questions

Answer:

WHERE filters rows. HAVING filters groups.

---

Q2. No primary key?

Answer:

Duplicate data and integrity issues.

---

Q3. Avoid duplicates?

Answer:

Use primary key and unique constraints.

---

Q4. DELETE vs TRUNCATE vs DROP?



Answer:

DELETE removes rows. TRUNCATE clears table. DROP removes table.

---

Q5. Normalization?

Answer:

Organizing data to reduce redundancy.

---

Q6. Index?

Answer:

Improves query performance.

## General Questions

---

Q7. Large data fetch?

Answer:

Use indexes and optimized queries.

---

Q8. Foreign key failure?

Answer:

Insertion/update fails.

---

### ● SQL – MUST-KNOW POINTS

Keys



Index

Normalization

Query optimization

---

### ◆ ☐ CODING LOGIC – INTERVIEW EXPLANATION (ALL LANGUAGES)

Reverse string

Logic: Swap characters from start and end.

Largest & second largest

Logic: Track two variables in one loop.

Palindrome

## General Questions

Logic: Compare string with reverse.

Duplicate removal

Logic: Use extra array / set.

Sum of digits

Logic: Modulo and division.

---

### ◆ ☐PROJECT DEEP-DIVE – SAFE ANSWERS

Architecture?

Answer:

Frontend → Logic → Database

Tech choice?

Answer:

Based on project requirement and ease.

Scaling users?

Answer:

Optimize database and logic.

Debug first?

Answer:

Logs and input data.

---

### ◆ ☐CORPORATE BEHAVIOUR – RIGHT ANSWERS

Code fails?

Fix, analyze, inform team.

## General Questions

Deadlines?

Prioritize and manage time.

Code review?

Accept feedback positively.

