



### PG-DAC

#### C++ Programming

1. How C++ manage its memory? Explain new & delete operators for variables, objects and arrays.
2. Is it possible to delete storage acquired by a local variable in static memory using delete keyword in C++?
3. What is the use of destructors? Write a legal example of a destructor.
4. What are the OOP concepts present in C++/Java? Explain with one example each.
5. How to create an abstract class in C++? How to write a pure virtual function? Can we write a body of pure virtual function?
6. What is getter and setter? Why do we use that? Write down a small code example.
7. Create a spiral matrix. (You can find this problem in Leetcode  
<https://leetcode.com/problems/spiral-matrix/>)
8. What is "this" pointer? Is it available for static, virtual, const and friend functions?
9. What is the need to write a user defined destructor? When should it be declared as "virtual"?
10. How does virtual function affect the size of an object? How are they executed at runtime?
11. What is the diamond problem? How to solve it?
12. What is shallow copy and deep copy? How is it implemented in C++? Explain with examples.
13. What is a smart pointer? Which are smart pointers in C++?
14. What is STL? Explain different components in STL with examples?

## Object Oriented Programming

1. What are object oriented concepts? What is difference between object-based, object-oriented and fully object-oriented language?
2. What are advantages of Object-Oriented Programming? What is data security?
3. What is class and object? Give real-life example.
4. What are characteristics of object? Explain them.
5. What is the need of getter and setter functions in class?
6. What is abstraction and encapsulation. Give real-life example.
7. What is polymorphism? What are its types? Explain them with examples.
8. What is method overloading? Which are the rules of method overloading? Why return type is not considered in method overloading?
9. What are different types of hierarchy? When to use which one?
10. What is the difference between method overloading and method overriding?
11. What is object slicing? Explain object slicing in context of up-casting?
12. What is down-casting and when it is required? Explain with code.
13. What do you know about association, composition and aggregation? Explain with the help of example.
14. What are different types of inheritance? Explain with the help of example. What are problems with multiple inheritance?
15. What is difference between interface, abstract class and non-abstract class? Which one to use where?
16. Which are the different types of design pattern? Explain singleton design pattern.

## Java Programming

1. What is a collection framework? Explain any 2 classes in Java Collections.
2. What is functional programming? Explain stream programming with examples.
3. Explain multithreading using examples. Explain standard examples like a producer-consumer problem.
4. How to create new threads? Explain "implements Runnable" vs "extends Thread".
5. What is Executor Framework?
6. start() vs run() in threads. What happens if you call run() instead of start() for an object of class that extends Thread?
7. What is the use of Garbage Collector in Java language?
8. Can you overload the main() method?
9. How to create thread-safe singleton pattern?
10. Can you override the static method?
11. Basic OOP principal. Encapsulation, Abstraction explained with good examples.
12. If multiple interfaces have the same method declaration, does it cause name collision or any other problem when they are implemented in a class?
13. What is Upcasting and Downcasting in Java? Where is it used in Java? Explain with examples.
14. Explain default methods in the interfaces.
15. How do HashSet and HashMap work internally in Java? What is the role of equals() and hashCode() methods in this context?
16. What is their initial capacity of a collection and what happens when it is exhausted? Explain with examples.
17. Is HashMap thread safe? How to make it thread safe?
18. Tell me the Difference between StringBuilder and StringBuffer. Why is String immutable?  
What is a String pool?
19. What is the difference between == and equals() in Java? Explain in context of primitive values and objects with examples.
20. What is a fail-fast and fail-safe iterator? Explain with code.
21. What is the lambda expression in Java? How data types of lambda arguments and return types are determined?
22. What is a functional interface? Which functional interfaces are predefined and where they are used?

1. What is time complexity and space complexity? Explain best case, worst case and average case time complexity with examples.
2. Display a given range of numbers in reverse order (using recursion).
3. How to reverse a singly linear linked list?
4. How to reverse a singly linear linked list using recursion?
5. Display singly linear linked list in reverse order.
6. Find the middle node of a singly linear linked list.
7. Check if a singly linear list contains a loop.
8. Check if the given singly linear list is palindrome or not.
9. Create a stack (LIFO) using queue (FIFO).
10. Implement a quick sort algorithm.
11. How to solve a rat in a maze problem?
12. What is a hash table? Implement hash tables with separate chaining.
13. Input a string from the user and find the character repeated maximum number of times (irrespective of case).
14. Print the length of the highest continuous number range in the given array.  
[\(https://www.geeksforgeeks.org/longest-consecutive-subsequence/\)](https://www.geeksforgeeks.org/longest-consecutive-subsequence/)
15. Flatten a multi level doubly linkedlist (You can find this problem in Leetcode -  
<https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list/>)
16. Find the maximum sum for any contiguous nodes in a given linkedlist.  
[\(https://www.geeksforgeeks.org/maximum-sum-contiguous-nodes-in-the-given-linked-list/\)](https://www.geeksforgeeks.org/maximum-sum-contiguous-nodes-in-the-given-linked-list/)
17. Given an array arr[] of N integers, calculate the median.  
[\(https://leetcode.com/problems/sliding-window-median/description/\)](https://leetcode.com/problems/sliding-window-median/description/)
18. Count number of palindrome trees in given input.  
[\(https://www.hackerearth.com/problem/algorithm/palindrome-tree/\)](https://www.hackerearth.com/problem/algorithm/palindrome-tree/)

## Advanced Java

1. What all servers/servlet containers are you aware of? Do you know about Jetty?
2. What is REST api? Explain in detail. What is @RestController?
3. Write a REST program from scratch in SpringBoot.
4. What is the significance of @CrossOrigin? How does it work?
5. How do @ResponseBody and @RequestBody work? What is ResponseEntity?
6. What is the difference between DTO and Entities? What are the advantages of using DTO?  
How have you implemented DTOs in your project?
7. How to secure Spring REST API? How authentication and authorization works?
8. What is Spring Boot? What do you mean by opinionated defaults? How does auto-configuration work?
9. What is the difference between @Component and @Bean? What will happen if I annotate an empty class with @Component?
10. What is IoC and Dependency Injection? What are different types?
11. What is the use of @Autowired? How autowiring resolves the field?
12. What is Maven? Explain role of pom.xml. Which dependencies have you used (in your project)?
13. What is SQL injection? How to prevent it in JDBC? Explain with code.
14. Explain Spring Data architecture. How is it related to Hibernate?
15. Explain hibernate/JPA entity life cycle?
16. How have you implemented many-to-many relations in your project? Explain code.
17. How can we call stored procedures in JDBC, hibernate, and Spring Data JPA?
18. What is the use of @Transactional? Why should it be used on the service layer?

## Database Technologies

1. What is normalization? What is its need? Explain 1NF, 2NF, 3NF and BCNF in detail.
2. Explain difference between super key, candidate key, alternate key, compound key, surrogate key, primary key, foreign key, and unique key.
3. What is difference between stored procedure and function?
4. Explain OUT parameter and IN-OUT parameter with an example (code).
5. What is Trigger? What are its application? Explain with example (code).
6. What is use of views? How to limit DML operations on views to the given criteria?
7. What are different types of views? What are applications of views?
8. What are advantages and limitations of indexes? What are different types of indexes? Explain with examples.
9. What is transaction? Explain ACID properties in RDBMS.
10. Find employees with third highest salary. Explain different ways to implement.
11. Find employees with salary more than their manager's salary.
12. What is difference between UNION and UNION ALL?
13. Change all gender values in the given table.
14. What will be output of different joins – Inner Join, Left Join, Right Join, Full Join, Cross Join, and Self Join.

```
CREATE TABLE t1 (c1 CHAR(1));
```

```
CREATE TABLE t2 (c2 CHAR(1));
```

```
INSERT INTO t1 VALUES ('A'), ('B'), ('B'), ('C'), ('P'), ('Q');
```

```
INSERT INTO t2 VALUES ('A'), ('B'), ('X'), ('Y'), ('Y'), ('Q');
```

15. Print department name and its average salary in descending order.
16. Display the employees, who are not managers.
17. How NoSQL is different than RDBMS? Where NoSQL is preferred over RDBMS?
18. Write a query to insert employee records in a Mongo collection.
19. Display name and salaries of employees in descending order for employees with salary more than 1000 (from a Mongo collection).
20. What is Big Data? How Big Data is different than traditional databases?

## **Web Programming**

1. What is HTTP protocol? What are contents of request and response?
2. Why HTTP is called as connection-less protocol? Why it is called as stateless protocol?
3. What is state management? Explain server side and client side state management objects.
4. Which are HTTP request methods? Explain difference between GET and POST methods.
5. What is difference between HTTP and HTTPS? What is SSL?
6. What is difference between HTML4 and HTML5?
7. What is difference between CSS, SCSS, and SASS?
8. What do you mean by responsive web UI? What is grid system in bootstrap?
9. What is DOM? Explain in context of HTML page.
10. What is difference between local storage and session storage? How to use them in JS code?
11. What is difference between == and === operator in JS?
12. What are different ways of writing functions in JS?
13. What is jQuery? What is advantage?
14. What is AJAX? How to use it with and without jQuery?
15. What are different ways to call REST APIs from JS? Write code.
16. What is closure in JS?
17. What is the difference between client-side and server-side JavaScript?
18. Write a Student class in JS. How to use it?

## MERN Stack

1. Explain asynchronous behavior of Node. What is callback functions?
2. How to create Http Server in NodeJS and handle different http methods?
3. How to read files using File System module? Explain blocking & non-blocking methods.
4. What is npm? Explain node modules? How they are implemented?
5. Why Node is light-weight compared to other web frameworks?
6. What are the core features of an Express framework?
7. Explain Middleware. How to setup express application? Explain Routing.
8. How to allow CORS in Express?
9. Write a code for executing DML and DQL queries in NodeJS.
10. What is react? How it differs from Angular?
11. What is virtual DOM? How it works? What are its advantages?
12. What is difference between react state and props? Give example code snippets.
13. How to send data from parent component to the child component? How to send notification from child component to the parent component?
14. What is react hook? Explain the useLocation(), and useNavigate() with code examples.
15. Explain syntax (arguments) of useEffect(). Explain a scenario where you have used its second argument.
16. What is JSX? How it works?
17. How to render a list of products in react? What is significance of key?
18. What is react router? How you have used it in your project? Explain with code snippet.
19. How to consume REST APIs in react?
20. How to implement security in MERN application using JWT?

## **MS.NET**

1. Which are the components of .net framework?
2. What is assembly? What is manifest?
3. What is GAC? How it handles DLL hell problem?
4. What is the difference between sealed override and abstract override?
5. What is delegate? Explain the types of delegate?
6. What the difference is between finalize and dispose? Where using keyword is used?
7. Explain ASP.NET architecture with IIS7.
8. Explain ADO.NET entity framework.
9. Explain MVC application life cycle.
10. Explain what is routing in MVC? What are the three segments for routing important? What is default route?
11. What are Filters in MVC? Explain action filters.
12. What are the methods of handling an Error in MVC?
13. Differences between Razor and ASPX View Engine in MVC?
14. Explain .NET core features and architecture.
15. How to develop REST services in .NET?

## **Operating Systems**

1. What is OS? What are its important functions?
2. What is system call? How it is executed?
3. Explain terms: Multi-programming, Multi-tasking, Multi-threading, Multi-processing and Multi-user?
4. Explain process life cycle.
5. What are Linux IPC mechanisms? Which one is fastest? How it works?
6. What is difference between process and thread?
7. What is difference between semaphore and mutex?
8. What is file? What is Filesystem?
9. What is paging? What is page fault and how it is handled?
10. What is difference between starvation and deadlock?
11. Explain OS booting.

## **Linux Programming**

12. Explain Linux booting.
13. How many Linux run-levels are there? Which features are enabled in each runlevel?
14. Advantages of Linux. What do you mean by open source?
15. Which Linux distributions you have used? What is major difference?
16. What is shell? Types of shells in Linux?
17. How to rename a file in Linux? How to hide a file?
18. How does `tee` command work? Explain with example.
19. Write a command to find the file from directory?
20. Write a command to find the file which contains the given word from the given directory?
21. What is difference between |, &&, and || on Linux command line?
22. How to redirect `stderr` and `stdout`?
23. What is SSH? How it works? How to use SSH without password?
24. What is the execution order of `~/.bashrc`, `~/.bash\_profile` and `~/.bash\_login`?
25. What is shell scripting? Does shell scripting use compiler? How is shell scripting executed?
26. What is significance of shebang?
27. How to execute one shell script inside another shell script?
28. Write Linux command to check background process? How you can kill multiple processes of the same program?
29. What are environment variables in Linux?
30. What are default file permissions in Linux? How to change them?

## **Software Development Methodologies**

1. What are the differences between Agile and traditional project management (Waterfall)?
2. What is scrum? What are the roles in Scrum? Explain their responsibilities.
3. Explain the terms story, task, epic and sprint. What is their relation?
4. What is cloud computing? Explain cloud service models: IaaS, PaaS, SaaS, FaaS, DaaS.
5. How to create EC2 instance in AWS?
6. What are the differences between vertical and horizontal scaling? Which one would you prefer and why?
7. What is use of these AWS services – EC2, Route53, S3, RDS, Beanstalk, Amplify?
8. Which are most popular DevOps tools (mention at least 5)? What is their use?
9. What is DevOps? Explain DevOps life cycle.
10. What are the differences between DevOps and agile methodologies?
11. What is micro-service? How is its advantage over monolithic application?
12. What is GIT? Explain GIT workflow.
13. What is GIT branching? How to implement it? Write commands to create branch, merge branch, and delete branch.
14. What is Docker? Explain Docker architecture.
15. What is Docker image and container? Explain life cycle of Docker container.
16. An application has three components i.e. MySQL database, Spring Boot RES services, and React front end. Explain steps to containerize this application.
17. What is orchestration? Which tools can be used for the same.
18. What is Kubernetes? Explain Kubernetes architecture.
19. What is pod? How do you scale pods?
20. What is CI? What is CD? Which are popular tools for CI/CD pipeline?
21. What is Jenkins? Explain Jenkins architecture.
22. How would you create pipeline in Jenkins?

## **Project Interview Questions**

1. Can you explain your C-DAC project? Draw block diagrams as appropriate.
2. Explain how OOPs concepts are implemented in your project?
3. Draw use-case diagrams, class diagrams and ER diagrams of your project?
4. Explain the n-tier architecture of your project?
5. Which advanced features have you used in your project?
6. What was your role in your project and explain what you did in it?
7. Which software development methodology (model/architecture) have you used in your project? Explain its process.
8. Which Design patterns are used in your project?
9. What are the difficulties you have faced during this project and How have you overcome it?
10. Which database is used in your Project? Why? Explain database design.
11. Explain the data access layer of your database with code.
12. Have you used AJAX in your project? How?
13. Which frontend technology is used in your project?
14. Explain configuration files used in your project?
15. Explain security implementation of your project?
16. How to build a CI/CD pipeline for the project?
17. Are you aware of Separation of Concerns (SoC) in designing an application?
18. How did you implement session management in your project?
19. What is the difference between authentication and authorisation?
20. Explain the authentication flow of your project. Did you use JWT? If not, why? If yes, why and how does it work?
21. How authorization is implemented in your project? Which roles are present in your project? How are resources mapped to the roles? Explain with relevant code.
22. How to host an application online? Did you try to host your project online?
23. What is DNS? Explain how a request to "www.google.co.in" ends up at google servers in overview.
24. What is scaling? Explain strategies for scaling the project.
25. What are micro-services? Have you implemented them in your project?

## **HR Interview Questions**

1. Tell us something about yourself.
2. Tell us some of your strengths and weaknesses. How you can overcome your weaknesses?
3. Are you comfortable working in a team? What makes you think so?
4. If you were hiring for this position, what qualities would you look for in a potential candidate?
5. How would you rate yourself as a candidate on a scale of 1-10? Give Reasons.
6. What is your career objective?
7. What motivates you at work?
8. What is your opinion about working in night shifts or over the weekends?
9. Do you think you have made the right career choice? Why?
10. Discuss the most stressful situation you came across in life.
11. Why should we hire you?
12. What is your idea of an ideal company?
13. Tell us something about your achievements during your education.
14. Do you think your marks in academics truly reflect your potential?
15. Tell us something about your hobbies.
16. How do you see yourself 5 years from now?
17. Tell us about your dream job.
18. Is there any particular kind of person you cannot work with?
19. Tell us what do you know about our company?
20. How long do you think you can work for our company?
21. Are you willing to sign a service bond with our company? If yes for what duration? If no, why not?
22. How important would the location of your job be for you? Why?
23. Which is the subject that you have liked in your studies till date? Why?
24. What are the things that might make you leave a job?
25. We are hiring for test engineer position. Are you willing to join? Why?
26. What is the reason for the gap in your education? (Answer only if applicable to you)
27. Why is your academic performance in X, XII & Graduation not consistent? (Answer only if applicable to you)?
28. There is a 2/3/4-year gap after finishing your education. Why? (Answer only if applicable to you).
29. Your graduation branch is different. Why do you want to change your line by taking this job? (Answer only if applicable to you).
30. Your previous academic performance has been below average. What makes you think your work performance will be better? (Answer only if applicable to you).