Cognizant Interview Experience

C# Programming Questions

1. Write a C# program to find the maximum occurrence of a character in a given string.
2. String C# Question: Find the occurrence of letter which is repeated maximum number of times (consider uppercase as lowercase).

Java Context Questions

3. Inheritance
4. Inheritance in C# (Multiple inheritance support via interfaces)
5. Static Methods & Variables
6. Access Modifiers

Database Questions

7. Indexes in SQL
8. How to Optimize the Database
9. ACID Properties
10. NoSQL vs SQL Databases Differences
11. Normalization
12. Join Tables and Find Same Data
13. SQL Subsequence Function

Web Development Task

14. Create User Login Password Field Using HTML, CSS

Data Structures

15. Linked List and Array Questions

Personal Questions

16. Resume-Based Questions

17. Project-Related Questions (Why/What type questions)

18. Internship Experience Questions

19. "Why C# when you don't have knowledge" Question

20. Certification Course Questions

21. Hobbies Question

22. Riddle: 2 Liars and 1 Honest Person - How to Choose the Right Box

23. Basic HR Questions on Relocation and Shift Work

Technical Interview Questions and Expected Answers

C# Programming Question

1. Write a C# program to find the maximum occurrence of a character in a given string.

Expected Answer:

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("Enter a string: ");
        string input = Console.ReadLine();

        char maxChar = FindMaxOccurringChar(input);

        Console.WriteLine($"The character with maximum occurrence is '{maxChar}'");
    }

    static char FindMaxOccurringChar(string str)
    {
        // Create a dictionary to store character counts
        Dictionary<char, int> charCount = new Dictionary<char, int>();

        // Count occurrences of each character
        foreach (char c in str)
        {
            if (charCount.ContainsKey(c))
                charCount[c]++;
            else
                charCount[c] = 1;
        }

        // Find the character with maximum count
        char maxChar = ' ';
        int maxCount = 0;

        foreach (var pair in charCount)
        {
            if (pair.Value > maxCount)
            {
                maxCount = pair.Value;
                maxChar = pair.Key;
            }
        }

        return maxChar;
    }
}
```

Java Context Questions

2. Inheritance

Expected Answer: Inheritance is an OOP principle where a class (subclass/derived class) inherits properties and methods from another class (superclass/base class). Java supports single class inheritance but multiple interface inheritance. The "extends" keyword is used to implement inheritance.

```
// Base class
class Animal {
    protected String name;

    public void eat() {
        System.out.println(name + " is eating");
    }
}

// Derived class
class Dog extends Animal {
    public Dog(String name) {
        this.name = name;
    }

    public void bark() {
        System.out.println(name + " is barking");
    }
}
```

3. Static Methods & Variables

Expected Answer: Static members belong to the class itself rather than instances. Static variables are shared across all instances and are initialized only once at the start of program execution. Static methods belong to the class and can be called without creating an instance.

```
public class Counter {
    // Static variable
    public static int count = 0;

    // Instance variable
    public int instanceCount = 0;

    // Constructor
    public Counter() {
        count++;
        instanceCount++;
    }

    // Static method
    public static void displayCount() {
        System.out.println("Total count: " + count);
    }
}
```

4. Access Modifiers

Expected Answer: Java has four access modifiers:

- **public**: Accessible from any class
- **protected**: Accessible within the same package and subclasses
- **default** (no modifier): Accessible only within the same package
- **private**: Accessible only within the same class

These control the visibility and accessibility of classes, methods, and variables.

5. Indexes in SQL

Expected Answer: Indexes are special database structures that improve query performance. They work similar to a book's index by allowing the database to find data without scanning the entire table.

Types:

- **Clustered Index**: Determines physical order of data (only one per table)
- **Non-clustered Index**: Stores logical order with pointers to physical data
- **Composite Index**: Index on multiple columns
- **Unique Index**: Ensures uniqueness of the indexed column values

When to use indexes:

- Columns used in WHERE clauses
- Columns used in JOIN conditions
- Columns used in ORDER BY or GROUP BY operations

6. How to Optimize the Database

Expected Answer:

- Proper indexing on frequently queried columns
- Normalization to reduce data redundancy
- Query optimization (avoiding SELECT *, using proper JOINs)
- Partitioning large tables
- Regular database maintenance (updating statistics, rebuilding indexes)
- Using stored procedures for complex operations
- Implementing caching mechanisms
- Hardware optimization (SSD storage, sufficient RAM)
- Proper database configuration settings
- Using execution plans to identify bottlenecks

7. Join Tables and Find Same Data

Expected Answer:

```
-- Using INNER JOIN to find matching records
SELECT a.column1, a.column2, b.column1, b.column2
FROM TableA a
INNER JOIN TableB b ON a.key_column = b.key_column
WHERE a.some_column = b.some_column;

-- Finding duplicate records in a single table
SELECT column1, column2, COUNT(*)
FROM Table
GROUP BY column1, column2
HAVING COUNT(*) > 1;
```

Additional Interview Questions

8. Web Development Task - Create User Login Password Field

Expected Answer:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .login-container {
            width: 300px;
            margin: 0 auto;
            padding: 20px;
            background-color: #f5f5f5;
            border-radius: 5px;
        }

        .password-field {
            width: 100%;
            padding: 10px;
            margin: 8px 0;
            background-color: #e8f0fe;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-sizing: border-box;
        }

        .login-button {
            width: 100%;
            padding: 10px;
            background-color: #4285f4;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h2>User Login</h2>
        <form>
            <label for="username">Username:</label>
            <input type="text" id="username" name="username"><br>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" class="password-field"><br>

            <input type="submit" value="Login" class="login-button">
        </form>
```

```
    </div>
</body>
</html>
```

9. Linked List and Array Questions

Expected Answer:Arrays:

- Fixed size (in Java/C#)
- Continuous memory allocation
- Direct access via index (O(1))
- Efficient for reading but inefficient for insertion/deletion in the middle

Linked Lists:

- Dynamic size
- Non-continuous memory allocation
- Sequential access (O(n))
- Efficient for insertion/deletion but inefficient for random access

Implementation differences:

- Arrays require size declaration at creation (in Java/C#)
- Linked lists grow dynamically as elements are added
- Arrays have better cache locality for better performance in iterations

10. SQL Queries - JOIN and AVG

Expected Answer:

```
-- Basic JOIN query
SELECT e.employee_id, e.name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id;

-- Using AVG function
SELECT d.department_name, AVG(e.salary) as average_salary
FROM employees e
JOIN departments d ON e.department_id = d.department_id
GROUP BY d.department_name;

-- Using LIKE with JOIN
SELECT e.employee_id, e.name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id
WHERE e.name LIKE 'J%';
```

11. Resume-Based Questions

Expected Answer: Be prepared to discuss all projects listed on your resume in detail. Demonstrate technical knowledge of the technologies used, explain specific challenges faced and how they were resolved. Clearly articulate your personal role and contributions to each project. Draw connections between past experiences and the position you're applying for. Use the STAR method (Situation, Task, Action, Result) to structure your responses.

12. Certification Course Questions

Expected Answer: Review core concepts and practical applications from your certification courses before the interview. Be ready to explain how these certifications relate to real-world scenarios and the specific role you're applying for. Prepare examples of how you've applied certification knowledge in practical situations. Focus on both theoretical understanding and hands-on implementation of the concepts covered in your certifications.

13. Hobbies Question

Expected Answer: Discuss your hobbies with genuine enthusiasm and authenticity. Consider how your personal interests might demonstrate valuable skills like problem-solving, creativity, persistence, or teamwork. For technical hobbies, be ready to explain how they've enhanced your professional skills. For non-technical hobbies, highlight transferable skills or how they provide balance to your technical work. Use this opportunity to show your personality and potential cultural fit with the organization.

Technical Interview Questions and Expected Answers

C# Programming Questions

1. Write a C# program to find the maximum occurrence of a character in a given string.

Expected Answer:

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("Enter a string: ");
        string input = Console.ReadLine();

        char maxChar = FindMaxOccurringChar(input);

        Console.WriteLine($"The character with maximum occurrence is '{maxChar}'");
    }

    static char FindMaxOccurringChar(string str)
    {
        // Create a dictionary to store character counts
        Dictionary<char, int> charCount = new Dictionary<char, int>();

        // Count occurrences of each character
        foreach (char c in str)
        {
            if (charCount.ContainsKey(c))
                charCount[c]++;
            else
                charCount[c] = 1;
        }

        // Find the character with maximum count
        char maxChar = ' ';
        int maxCount = 0;

        foreach (var pair in charCount)
        {
            if (pair.Value > maxCount)
            {
                maxCount = pair.Value;
                maxChar = pair.Key;
            }
        }

        return maxChar;
    }
}
```

2. String C# Question: Find the occurrence of letter which is repeated maximum number of times (consider uppercase as lowercase)

Expected Answer:

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("Enter a string: ");
        string input = Console.ReadLine();

        char maxChar = FindMaxOccurringCharCaseInsensitive(input);

        Console.WriteLine($"The character with maximum occurrence is '{maxChar}'");
    }

    static char FindMaxOccurringCharCaseInsensitive(string str)
    {
        // Create a dictionary to store character counts
        Dictionary<char, int> charCount = new Dictionary<char, int>();

        // Convert string to lowercase and count occurrences
        string lowerStr = str.ToLower();

        foreach (char c in lowerStr)
        {
            if (charCount.ContainsKey(c))
                charCount[c]++;
            else
                charCount[c] = 1;
        }

        // Find the character with maximum count
        char maxChar = ' ';
        int maxCount = 0;

        foreach (var pair in charCount)
        {
            if (pair.Value > maxCount)
            {
                maxCount = pair.Value;
                maxChar = pair.Key;
            }
        }

        return maxChar;
```

```
    }
}
```

Java Context Questions

3. Inheritance

Expected Answer: Inheritance is an OOP principle where a class (subclass/derived class) inherits properties and methods from another class (superclass/base class). Java supports single class inheritance but multiple interface inheritance. The "extends" keyword is used to implement inheritance.

```
// Base class
class Animal {
    protected String name;

    public void eat() {
        System.out.println(name + " is eating");
    }
}

// Derived class
class Dog extends Animal {
    public Dog(String name) {
        this.name = name;
    }

    public void bark() {
        System.out.println(name + " is barking");
    }
}
```

4. Inheritance in C# (Multiple inheritance support via interfaces)

Expected Answer: C# supports single class inheritance but allows multiple interface inheritance. This enables a class to implement functionality from multiple sources while avoiding the "diamond problem" that can occur with multiple class inheritance.

```
// Base class
public class Vehicle
{
    public void Start()
    {
        Console.WriteLine("Vehicle started");
    }
}

// Interface 1
public interface IElectric
{
    void Charge();
}

// Interface 2
public interface IAutonomous
{
    void SelfDrive();
}

// Class implementing multiple inheritance via interfaces
public class ElectricCar : Vehicle, IElectric, IAutonomous
{
    public void Charge()
    {
        Console.WriteLine("Car is charging");
    }

    public void SelfDrive()
    {
        Console.WriteLine("Car is self-driving");
    }
}
```

5. Static Methods & Variables

Expected Answer: Static members belong to the class itself rather than instances. Static variables are shared across all instances and are initialized only once at the start of program execution. Static methods belong to the class and can be called without creating an instance.

```
public class Counter {
    // Static variable
    public static int count = 0;

    // Instance variable
    public int instanceCount = 0;

    // Constructor
    public Counter() {
        count++;
        instanceCount++;
    }

    // Static method
    public static void displayCount() {
        System.out.println("Total count: " + count);
    }
}
```

6. Access Modifiers

Expected Answer: Java has four access modifiers:

- **public**: Accessible from any class
- **protected**: Accessible within the same package and subclasses
- **default** (no modifier): Accessible only within the same package
- **private**: Accessible only within the same class

These control the visibility and accessibility of classes, methods, and variables.

Database Questions

7. Indexes in SQL

Expected Answer: Indexes are special database structures that improve query performance. They work similar to a book's index by allowing the database to find data without scanning the entire table.

Types:

- **Clustered Index**: Determines physical order of data (only one per table)
- **Non-clustered Index**: Stores logical order with pointers to physical data
- **Composite Index**: Index on multiple columns
- **Unique Index**: Ensures uniqueness of the indexed column values

When to use indexes:

- Columns used in WHERE clauses
- Columns used in JOIN conditions
- Columns used in ORDER BY or GROUP BY operations

8. How to Optimize the Database

Expected Answer:

- Proper indexing on frequently queried columns
- Normalization to reduce data redundancy
- Query optimization (avoiding SELECT *, using proper JOINs)
- Partitioning large tables
- Regular database maintenance (updating statistics, rebuilding indexes)
- Using stored procedures for complex operations
- Implementing caching mechanisms
- Hardware optimization (SSD storage, sufficient RAM)
- Proper database configuration settings
- Using execution plans to identify bottlenecks

9. ACID Properties

Expected Answer: ACID is an acronym representing the four key properties that guarantee reliable database transactions:

- **Atomicity**: Ensures that a transaction is treated as a single, indivisible unit. Either all operations in the transaction are completed successfully, or none are (rollback).
- **Consistency**: Ensures that a transaction brings the database from one valid state to another valid state, maintaining all predefined rules and constraints.
- **Isolation**: Ensures that concurrent transactions do not interfere with each other. Transactions execute as if they were running one after another, even if they are running simultaneously.
- **Durability**: Ensures that once a transaction is committed, it remains committed even in the case of system failure. The changes made by the transaction are permanently stored.

Example: When transferring money between accounts, ACID properties ensure the money is neither lost nor duplicated, regardless of system failures or concurrent operations.

10. NoSQL vs SQL Databases Differences

Expected Answer:SQL Databases:

- Structured data with predefined schema (tables, rows, columns)
- Strong ACID compliance
- Vertical scalability (scale-up with more powerful hardware)
- Use standardized SQL language
- Examples: MySQL, PostgreSQL, Oracle, SQL Server

NoSQL Databases:

- Schema-less or flexible schema design
- Horizontal scalability (scale-out across multiple servers)
- Eventual consistency over strict ACID (though some offer ACID)
- Optimized for specific data models:
    - Document stores (MongoDB, CouchDB)
    - Key-value stores (Redis, DynamoDB)
    - Column-family stores (Cassandra, HBase)
    - Graph databases (Neo4j, ArangoDB)

When to use SQL:

- Complex queries and transactions
- Data integrity is critical
- Structured data with stable schemas

When to use NoSQL:

- Handling large volumes of unstructured/semi-structured data
- Rapid development with evolving data models
- Distributed database architecture
- High throughput with simpler queries

11. Normalization

Expected Answer: Normalization is the process of organizing database tables to minimize redundancy and dependency by dividing large tables into smaller, related tables.

1NF (First Normal Form):

- Each table cell should contain a single value
- Each record needs to be unique No
- repeating groups of columns

2NF (Second Normal Form):

- Must be in 1NF
- All non-key attributes are fully functionally dependent on the primary key

3NF (Third Normal Form):

- Must be in 2NF
- No transitive dependencies (non-key attributes depend only on the primary key)

BCNF (Boyce-Codd Normal Form):

- A stronger version of 3NF
- For any dependency A → B, A should be a super key

Benefits of Normalization:

- Reduces data redundancy

- Improves data integrity
- Better database organization
- More efficient updates

12. Join Tables and Find Same Data

Expected Answer:

```
-- Using INNER JOIN to find matching records
SELECT a.column1, a.column2, b.column1, b.column2
FROM TableA a
INNER JOIN TableB b ON a.key_column = b.key_column
WHERE a.some_column = b.some_column;

-- Finding duplicate records in a single table
SELECT column1, column2, COUNT(*)
FROM Table
GROUP BY column1, column2
HAVING COUNT(*) > 1;
```

13. SQL Subsequence Function

Expected Answer: SQL doesn't have a built-in subsequence function, but we can implement it using string functions. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

To check if string B is a subsequence of string A:

```
CREATE FUNCTION IsSubsequence(@str VARCHAR(MAX), @subseq VARCHAR(MAX))
RETURNS BIT
AS
BEGIN
    DECLARE @i INT = 1;
    DECLARE @j INT = 1;
    DECLARE @str_len INT = LEN(@str);
    DECLARE @subseq_len INT = LEN(@subseq);

    WHILE @i <= @str_len AND @j <= @subseq_len
    BEGIN
        IF SUBSTRING(@str, @i, 1) = SUBSTRING(@subseq, @j, 1)
            SET @j = @j + 1;

        SET @i = @i + 1;
    END

    RETURN CASE WHEN @j > @subseq_len THEN 1 ELSE 0 END;
END;
```

Example usage:

```
SELECT dbo.IsSubsequence('database', 'dtbs'); -- Returns 1 (true)
```

Other rare SQL functions worth knowing:

- PIVOT/UNPIVOT
- STRING_AGG (SQL Server)
- LISTAGG (Oracle)
- JSON functions
- Window functions (ROW_NUMBER, RANK, DENSE_RANK, NTILE)

Web Development Task

14. Create User Login Password Field Using HTML, CSS

Expected Answer:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .login-container {
            width: 300px;
            margin: 0 auto;
            padding: 20px;
            background-color: #f5f5f5;
            border-radius: 5px;
        }

        .password-field {
            width: 100%;
            padding: 10px;
            margin: 8px 0;
            background-color: #e8f0fe;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-sizing: border-box;
        }

        .login-button {
            width: 100%;
            padding: 10px;
            background-color: #4285f4;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h2>User Login</h2>
        <form>
            <label for="username">Username:</label>
            <input type="text" id="username" name="username"><br>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" class="password-field"><br>

            <input type="submit" value="Login" class="login-button">
        </form>
```

```
    </div>
</body>
</html>
```

Data Structures

15. Linked List and Array Questions

Expected Answer:Arrays:

- Fixed size (in Java/C#)
- Continuous memory allocation
- Direct access via index (O(1))
- Efficient for reading but inefficient for insertion/deletion in the middle

Linked Lists:

- Dynamic size
- Non-continuous memory allocation
- Sequential access (O(n))
- Efficient for insertion/deletion but inefficient for random access

Implementation differences:

- Arrays require size declaration at creation (in Java/C#)
- Linked lists grow dynamically as elements are added
- Arrays have better cache locality for better performance in iterations

Personal Questions

16. Resume-Based Questions

Expected Answer: Be prepared to discuss all projects listed on your resume in detail. Demonstrate technical knowledge of the technologies used, explain specific challenges faced and how they were resolved. Clearly articulate your personal role and contributions to each project. Draw connections between past experiences and the position you're applying for. Use the STAR method (Situation, Task, Action, Result) to structure your responses.

17. Project-Related Questions (Why/What type questions)

Expected Answer: For "why" questions:

- Why did you choose a particular technology/framework? Focus on technical advantages, team expertise, scalability, or business requirements that drove the decision.
- Why did you structure the project in a certain way? Explain architectural decisions based on requirements, maintainability, and future scalability.

For "what" questions:

- What were your key learnings? Highlight both technical insights and soft skills like collaboration or problem-solving.
- What would you do differently? Show reflection and growth mindset by mentioning improvements you'd make with hindsight.
- What challenges did you face? Discuss technical hurdles and how you overcame them, demonstrating problem-solving abilities.

18. Internship Experience Questions

Expected Answer: Discuss your internship experience by highlighting:

- Specific projects and technologies you worked with
- Skills you developed during the internship
- Challenges you faced and how you overcame them
- How you collaborated with team members
- Feedback you received and how you implemented it
- How the experience shaped your career goals

Use concrete examples to demonstrate your growth and contributions. Explain how your internship experience prepared you for the role you're applying for.

19. "Why C# when you don't have knowledge" Question

Expected Answer: When asked why you chose C# despite limited knowledge:

- Express your interest in learning and mastering C# due to its versatility and widespread use in enterprise applications
- Highlight transferable programming skills from languages you already know
- Mention any steps you've already taken to learn C# (courses, self-study, small projects)
- Discuss C#'s benefits for the type of development you want to do (web, mobile, gaming with Unity, etc.)
- Demonstrate your ability to quickly learn new technologies based on past experiences
- Express willingness to invest extra time to get up to speed

20. Certification Course Questions

Expected Answer: Review core concepts and practical applications from your certification courses before the interview. Be ready to explain how these certifications relate to real-world scenarios and the specific role you're applying for. Prepare examples of how you've applied certification knowledge in practical situations. Focus on both theoretical understanding and hands-on implementation of the concepts covered in your certifications.

21. Hobbies Question

Expected Answer: Discuss your hobbies with genuine enthusiasm and authenticity. Consider how your personal interests might demonstrate valuable skills like problem-solving, creativity, persistence, or teamwork. For technical hobbies, be ready to explain how they've enhanced your professional skills. For non-technical hobbies, highlight transferable skills or how they provide balance to your technical work. Use this opportunity to show your personality and potential cultural fit with the organization.

22. Riddle: 2 Liars and 1 Honest Person - How to Choose the Right Box

Expected Answer: The classic setup is: There are three people guarding three boxes, one containing a treasure. One person always tells the truth, and two always lie. You don't know who is who. You can ask one question to one person to find the right box.

Solution: Ask any person, "If I were to ask the person to your right which box contains the treasure, what would they say?" Then choose the opposite of what they tell you.

Explanation:

- If you ask the truth-teller: They will correctly tell you what the liar would say, which is the wrong box.
- If you ask a liar: They will lie about what the other person would say. If the other person is a truth-teller, the liar will not tell you what the truth-teller would actually say (the correct box). If the other person is a liar, the liar will not tell you what that other liar would say (the wrong box).

In all cases, the answer you receive will identify a wrong box, so you should choose the opposite.

23. Basic HR Questions on Relocation and Shift Work

Expected Answer: For relocation questions:

- Be honest about your willingness and flexibility
- If open to relocation, express enthusiasm about the opportunity to experience a new location
- If there are constraints, explain them professionally while emphasizing your interest in the role
- Ask about relocation assistance if applicable

For shift work questions:

- Demonstrate flexibility while being honest about your preferences
- Discuss any previous experience with different shifts
- Inquire about rotation patterns if applicable
- Ask about advance notice for schedule changes
- Express understanding of business needs requiring 24/7 coverage in some roles