

Phase 1: C# Basics & Core Concepts - Comprehensive Notes

1. Variables & Data Types

C# is a statically typed language. Variables must be declared with a data type.

Value Types: int, float, bool, char - stored in stack memory.

Reference Types: string, arrays, class, List<T> - stored in heap memory.

Example:

```
int age = 25;
string name = "Alice";
```

2. Control Flow (if, switch)

Control flow structures help control the logic of program execution.

- if / else / else if
- switch statements
- ternary (?:)

Example:

```
if (x > 0) Console.WriteLine("Positive");
else Console.WriteLine("Negative");
```

```
switch (choice) {
    case 1: Console.WriteLine("One"); break;
    default: Console.WriteLine("Default"); break;
}
```

3. Loops

for, while, do-while, and foreach loops are supported.

Example:

```
for (int i = 0; i < 5; i++) Console.WriteLine(i);
```

```
while (x < 10) { x++; }
```

```
foreach (var item in list) Console.WriteLine(item);
```

4. Functions & Recursion

Functions are declared using return type, name, parameters.

Recursion is when a function calls itself.

Example:

```
int Sum(int a, int b) => a + b;
```

```
int Factorial(int n) {
    if (n == 0) return 1;
    return n * Factorial(n - 1);
}
```

Phase 1: C# Basics & Core Concepts - Comprehensive Notes

5. Arrays

Fixed-size collection of items of the same type.

Key Methods/Properties:

- Length
- Indexing
- Array.Sort(), Array.Reverse()

Practice: reverse array, rotate, max/min

6. Strings

Immutable sequence of characters.

Key Methods:

- Length, ToUpper(), ToLower()
- Substring(), Replace()
- IndexOf(), Contains()
- Split(), Trim()
- StartsWith(), EndsWith()

Practice: palindrome, anagram, vowel count

7. List<T>

Dynamic array. Resizable.

Key Methods:

- Add(), Remove(), Insert()
- Sort(), Reverse(), Contains()

Practice: merge lists, remove duplicates

8. Dictionary (HashMap)

Stores key-value pairs.

Key Methods:

- Add(), Remove(), ContainsKey(), TryGetValue()

Practice: char count, group anagrams, frequency count

9. HashSet<T>

Unordered collection of unique elements.

Key Methods:

- Add(), Remove(), Contains()

Phase 1: C# Basics & Core Concepts - Comprehensive Notes

Practice: find duplicates, unique values

10. Stack & Queue

Stack: LIFO

Queue: FIFO

Stack Methods: Push(), Pop(), Peek()

Queue Methods: Enqueue(), Dequeue(), Peek()

Practice: valid parentheses, reverse string, first non-repeating char

11. Custom Classes & OOP

OOP pillars: Encapsulation, Inheritance, Polymorphism, Abstraction

Define class using `class` keyword.

Use constructors, fields, properties, methods.

Example:

```
class Person {  
    public string Name { get; set; }  
    public Person(string name) { Name = name; }  
}
```

12. LINQ Essentials

Used to query collections in a SQL-like manner.

Key LINQ Methods:

- Filtering: Where(x => x > 5)
- Projection: Select(x => x * 2)
- Aggregation: Count(), Sum(), Min(), Max()
- Ordering: OrderBy(), OrderByDescending()
- Grouping: GroupBy()
- Set operations: Distinct(), Intersect(), Union()

Practice: find top elements, filter data, map and transform

Transition Practice Problems:

1. Reverse a string using loop
2. Count characters using Dictionary
3. Remove duplicates from List<int>
4. Use LINQ to find the longest word
5. Merge two sorted arrays
6. Validate parentheses using Stack