Aufgabe 1: Multiple Choice (25 Punkte)

Bitte kreuzen Sie die jeweils einzig richtige Antwort an (1 Punkt pro Frage).

1. Strings sind in C# "immutable" (unveränderlich). Was ist die direkte Konsequenz daraus?

- ☐ Wenn man zwei Strings verbindet (`a + b`), wird der ursprüngliche String `a` überschrieben.
- ☐ Strings können nicht in einer `if`-Bedingung verwendet werden.
- ☐ Jede Operation, die einen String "ändert" (z.B. Replace), gibt ein komplett neues String-Objekt zurück.
- ☐ Strings verbrauchen keinen Arbeitsspeicher.

2. Warum kann man einem Array in C# nicht einfach ein weiteres Element hinzufügen (z.B. ein 6. Element bei Größe 5)?

- ☐ Weil Arrays eine feste Größe haben, die bei der Erstellung festgelegt wird.
- ☐ Weil man dazu erst die Methode `.Unlock()` aufrufen muss.
- ☐ Weil Arrays nur gerade Zahlen als Index erlauben.
- ☐ Das ist falsch; Arrays können beliebig wachsen.

3. Wenn Sie eine Variable vom Typ `int` an eine Methode übergeben (Call by Value), was passiert dann?

- ☐ Die Methode arbeitet mit einer Kopie des Wertes; das Original bleibt unverändert.
- ☐ Die Methode arbeitet direkt auf dem Originalspeicherplatz.
- ☐ Der Wert wird beim Übergeben automatisch auf 0 gesetzt.

4. Sie haben ein Array `string[] namen = new string[10];`. Wie greifen Sie auf das letzte Element zu?

- ☐ `namen[10]`
- ☐ `namen[9]`
- ☐ `namen[namen.Length]`
- ☐ `namen[-1]`

5. Welchen Zweck erfüllen Enums (z.B. `enum Status { Offen, Bearbeitet, Erledigt }`)?

- ☐ Sie dienen dazu, Texte alphabetisch zu sortieren.
- ☐ Sie machen den Code lesbarer, indem sie Zahlenwerte durch sprechende Namen ersetzen.
- ☐ Sie sind notwendig, um Datenbankverbindungen herzustellen.
- ☐ Sie sparen Speicherplatz im Vergleich zu `int`.

6. Welcher Fehler tritt auf, wenn Sie versuchen, auf einen Index zuzugreifen, der außerhalb der Array-Grenzen liegt (z.B. Index 10 bei einem Array der Länge 8)?

- ☐ Das Array wird automatisch erweitert, um den Index zu unterstützen.
- ☐ Es wird eine IndexOutOfRangeException zur Laufzeit ausgelöst.
- ☐ Das Programm gibt automatisch den Wert `null` zurück.

7. Welches ist die korrekte Syntax für String-Interpolation, um die Variable `name` in einen Text einzufügen?

- ☐ `String.Format("Hallo {name}")`
- ☐ `$"Hallo {name}"`
- ☐ `@"Hallo [name]"`
- ☐ `#"Hallo (name)"`

8. Was versteht man unter Methodenüberladung (Overloading)?

- ☐ Das Ersetzen einer Methode in einer abgeleiteten Klasse.
- ☐ Eine Methode, die sich selbst aufruft.
- ☐ Methoden, die denselben Namen haben, aber unterschiedliche Parameterlisten verwenden.
- ☐ Eine Methode, die eine andere Methode innerhalb derselben Klasse aufruft.

9. Was passiert bei einer Rekursion ohne Abbruchbedingung?

- ☐ Der Computer schaltet sich aus.
- ☐ Es kommt zu einem Fehler (StackOverflow / Speicherüberlauf).
- ☐ Die Methode gibt `null` zurück.
- ☐ Der Compiler verhindert das Kompilieren.

10. Welche Syntax erstellt korrekt ein 2D-Array (z.B. ein Schachbrett)?

- ☐ `int[][] brett = new int[8][8];`
- ☐ `int brett = new int(8,8);`
- ☐ `int[,] brett = new int[8,8];`
- ☐ `2D<int> brett = new 2D<int>(8);`

11. Die Methode `text.IndexOf("x")` sucht nach "x". Was gibt sie zurück, wenn "x" nicht gefunden wird?

- ☐ `0`
- ☐ `false`
- ☐ `-1`
- ☐ `null`

12. In welchem Szenario ist eine `List<T>` besser geeignet als ein Array?

- ☐ Wenn die Anzahl der Elemente vorher genau bekannt ist.
- ☐ Wenn man extrem speichersparend arbeiten muss.
- ☐ Wenn Elemente dynamisch hinzugefügt oder entfernt werden müssen.
- ☐ Wenn man nur primitive Datentypen (int) speichern will.

13. Sie fügen den Wert "5" zweimal in ein `HashSet<int>` ein. Was passiert?

- ☐ Das Set enthält danach zweimal die 5.
- ☐ Das Programm stürzt ab (Exception).
- ☐ Der zweite Eintrag wird ignoriert; die 5 ist nur einmal vorhanden.

14. Sie wollen Autokennzeichen (Key, z.B. "B-XY 123") und den Halter (Value, z.B. "Müller") speichern. Was nutzen Sie?

- ☐ `List<string>`
- ☐ `Dictionary<string, string>`
- ☐ `string[,]`
- ☐ `HashSet<string>`

15. Ein Emoji oder Sonderzeichen kann aus mehreren `char` bestehen. Wie nennt man die Einheit, die der Nutzer als *ein* sichtbares Zeichen wahrnimmt?

- ☐ Bit
- ☐ String-Element
- ☐ Graphem
- ☐ ASCII-Code

16. Wann wird der Codeblock einer `do-while`-Schleife ausgeführt?

- ☐ Nur, wenn die Bedingung von Anfang an wahr ist.
- ☐ Mindestens einmal, danach abhängig von der Bedingung.
- ☐ Nur, wenn die Bedingung `false` ist.
- ☐ Gar nicht, sie ist veraltet.

Aufgabe 2: Zuordnungsaufgaben (15 Punkte)

Ordnen Sie die Elemente korrekt zu. Tragen Sie jeweils den korrekten Buchstaben in den Zeilen der Aufgaben ein.

1. Datenstrukturen wählen (5 Punkte)

Welche Datenstruktur ist am besten geeignet?

Problemstellung	Buchstabe	Beste Datenstruktur
1. Speichern von Kursteilnehmern, Reihenfolge egal, keine Doppelten erlaubt.		A. `double[]`
2. Übersetzungstabelle (Deutsch -> Englisch).		B. `List<string>`
3. Ein Spielfeld "Vier gewinnt" (6 Reihen, 7 Spalten).		C. `HashSet<string>`
4. Messwerte einer Wetterstation (feste Anzahl: 24 Stundenwerte).		D. `int[,]` (2D Array)
5. Eine Chat-Historie (neue Nachrichten kommen ständig hinzu).		E. `Dictionary<string, string>`

2. Operatoren & Logik (5 Punkte)

Verbinden Sie den Operator mit seiner Bedeutung.

Operator	Buchstabe	Bedeutung
1. `!=`		A. Logisches ODER (mindestens eins muss wahr sein).
2. `||`		B. Division.
3. `/`		C. Kleiner als.
4. `<`		D. Logisches UND.
5. `&&`		E. Prüft auf Ungleichheit.

3. Real-World Datentypen (5 Punkte)

Wählen Sie den sinnvollsten Datentyp.

Anwendungsfall	Buchstabe	Datentyp
1. Ein einzelner Buchstabe (z.B. Geschlecht 'M'/'W').		A. `bool`
2. Ist der Benutzer eingeloggt? (Ja/Nein).		B. `decimal`
3. Ein genauer Finanzbetrag (z.B. Kontostand).		C. `string`
4. Eine E-Mail-Adresse.		D. `char`
5. Anzahl der Einwohner einer Stadt (keine Kommazahlen).		E. `int`

Aufgabe 3: Code-Analyse (10 Punkte)

Analysieren Sie die Code-Schnipsel und notieren Sie die Ausgabe. 2 Punkte je richtiger Lösung.

1. String-Operationen

```csharp
string s = "Hallowelt";
string teil = s.Substring(0, 5).ToUpper();
Console.WriteLine(teil);
```

Ausgabe: ____

2. Schleifen

```csharp
int[] zahlen = { 2, 4, 6, 8 };
int ergebnis = 0;
foreach(int z in zahlen) {
    if(z > 4) {
        ergebnis += z;
    }
}
Console.WriteLine(ergebnis);
```

Ausgabe: ____

3. Methoden

```csharp
int Verdoppeln(int x) {
    return x * 2;
}

int a = 3;
int b = Verdoppeln(a);
Console.WriteLine(a + b);
```

Ausgabe: ____

4. Listen

```csharp
List<string> tiere = new List<string> { "Hund", "Katze" };
tiere.Add("Maus");
tiere.Remove("Hund");
Console.WriteLine(tiere[0]);
```

Ausgabe: ____

5. Sets

```csharp
HashSet<int> ids = new HashSet<int>();
ids.Add(10);
ids.Add(20);
ids.Add(10);
Console.WriteLine(ids.Count);
```

Ausgabe: ____

Aufgabe 4: Lückentext & Syntax (10 Punkte)

Vervollständigen Sie den Code an den markierten Stellen. 2 Punkte je richtiger Lösung.

1. Array initialisieren

Ein Array für Kommazahlen erstellen, das Platz für 5 Werte bietet.

```csharp
double[] werte = new _______________________;
```

2. Methode schreiben

Eine Methode `SagHallo`, die einen Namen (string) annimmt und nichts zurückgibt (`void`).

```csharp
_______________ SagHallo(_____________________) { ... }
```

3. String-Änderungsmethoden

Ersetzen sie im Text das Wort "Welt" durch "C#".

```csharp
string text = "Hallo Welt";
string result = text____________________________;
Console.WriteLine(result);
```

Erwartete Ausgabe: Hallo C#

4. Liste erstellen

Erstellen Sie eine neue Liste für ganze Zahlen.

```csharp
List<int> meineZahlen = new _____________________________;
```

5. Dictionary Zugriff

Geben Sie den Wert für den Schlüssel "Admin" aus.

```csharp
Dictionary<string, int> rollen = new Dictionary<string, int>();
rollen.Add("Admin", 1);
int level = ________________________;
```