Advance Programming Techniques (APT)

Lecture # 37

Ehtisham Rasheed

Department of Computer Science
University of Gurjat, Gujrat

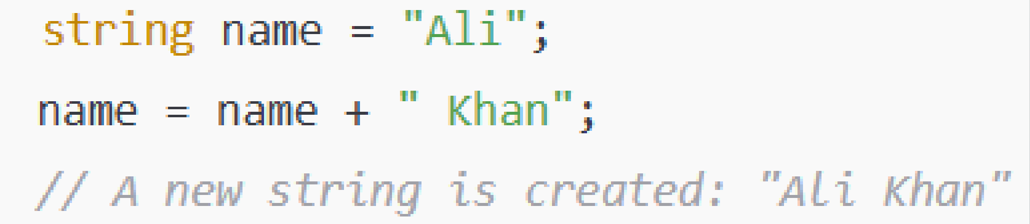
UNIVERSITY OF GUJRAT

Character

- In C# character is represented by char data type
- It stores a single 16-bit Unicode character and occupies 2 bytes of memory
- It must be declared with single quotes, such as char letter = 'A';
- The char type supports Unicode characters including letters, digits, symbols, and escape sequences like \n for newline
- A character constant is represented as an integer value (character code). E.g. 'z' corresponds to 122 and '\n' corresponds to 10.

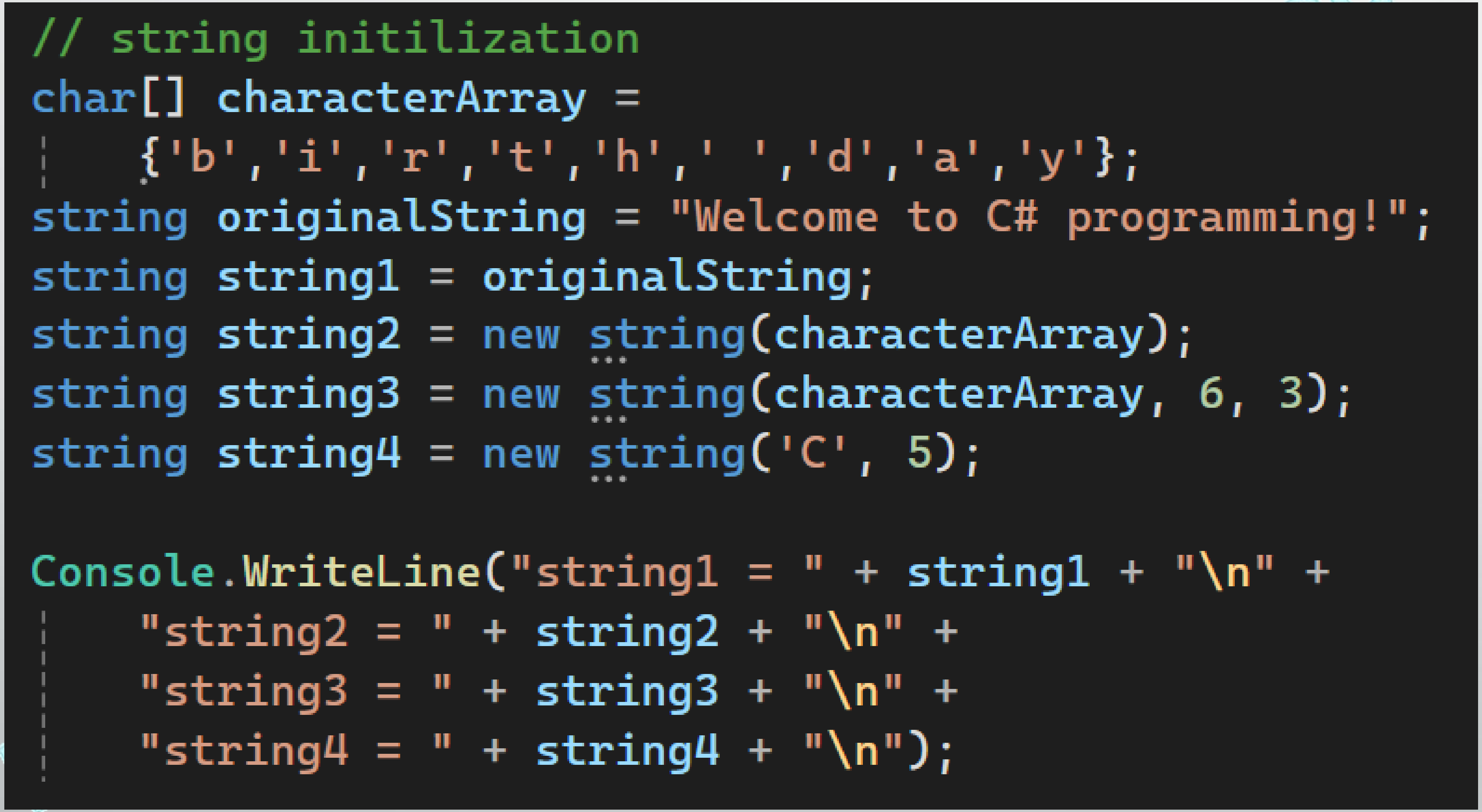
Strings

- A **string** is a sequence of characters
- Characters can be lowercase, uppercase, digits, special symbols etc
- Strings are declared with double quotes i.e. string color = “Blue”;
- Strings can be accessed like arrays using index.
- We can use foreach loop to iterate over characters.
- In C#, a string is **immutable**
- Meaning: once created, it **cannot be changed** — any modification creates a new string behind the scenes



```
string name = "Ali";
name = name + " Khan";
// A new string is created: "Ali Khan"
```

String Constructors



```
// string initilization
char[] characterArray =
    {'b','i','r','t','h',' ','d','a','y'};
string originalString = "Welcome to C# programming!";
string string1 = originalString;
string string2 = new string(characterArray);
string string3 = new string(characterArray, 6, 3);
string string4 = new string('C', 5);

Console.WriteLine("string1 = " + string1 + "\n" +
    "string2 = " + string2 + "\n" +
    "string3 = " + string3 + "\n" +
    "string4 = " + string4 + "\n");
```

Commonly Used String Functions

- Length
- ToUpper(), ToLower()
- Trim(), TrimStart(), TrimEnd()
- Substring()
- Contains(), StartWith(), EndWith()
- IndexOf(), LastIndexOf()
- Replace(), Split(), Join()
- Insert(), Remove()
- Compare(), CompareTo(), Equals(), Concat()
- IsNullOrEmpty() / IsNullOrWhiteSpace()

Verbatim Strings

- Verbatim strings contains multiline or path strings
- We use @ with verbatim strings
- Case sensitive and case insensitive comparison
- String Builder class

Character Functions

- Char.IsDigit()
- Char.IsLetter()
- Char.IsLetterOrDigit()
- Char.IsWhiteSpace()
- Char.IsUpper(), Char.IsLower()
- Char.ToUpper(), Char.ToLower()
- Char.IsSymbol()
- Char.IsPunctuation()

Real World Input Validation Examples

- Example 1: Check if user's name contains only letters
- Example 2: Validate numeric input
- Example 3: Validate email format (simple version)
- Example 4: Check for strong password validation