Féidearthachtaí as Cuimse
Infinite Possibilities

Week 10 - Tutorial

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Programming - Week 10

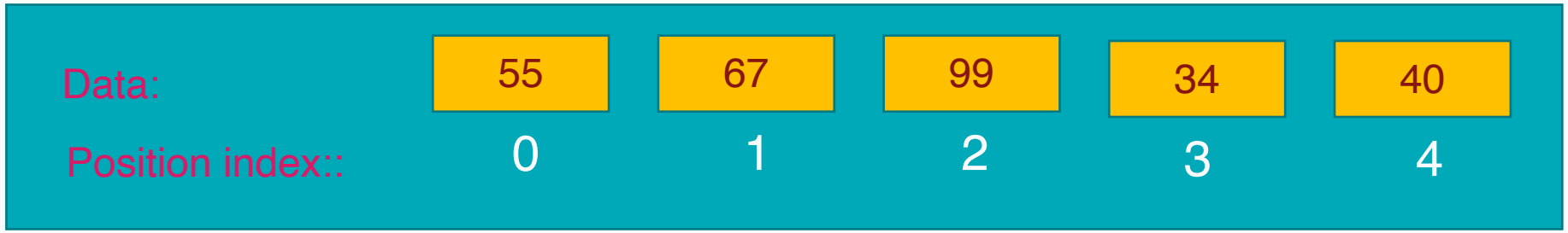
Overview

- Revision – Arrays
- Working with Arrays
- Pointers and Arrays
- Coding Example – Problem 1
- Mandatory Question
- Programming Pitfalls

Working with an Array

- An array is a linear data structure.
- Data stored in the array are sequential (one after the other).
- The data at the start of the array is in position 0. This is the index of the array. We can use the index to access the data stored in the array.

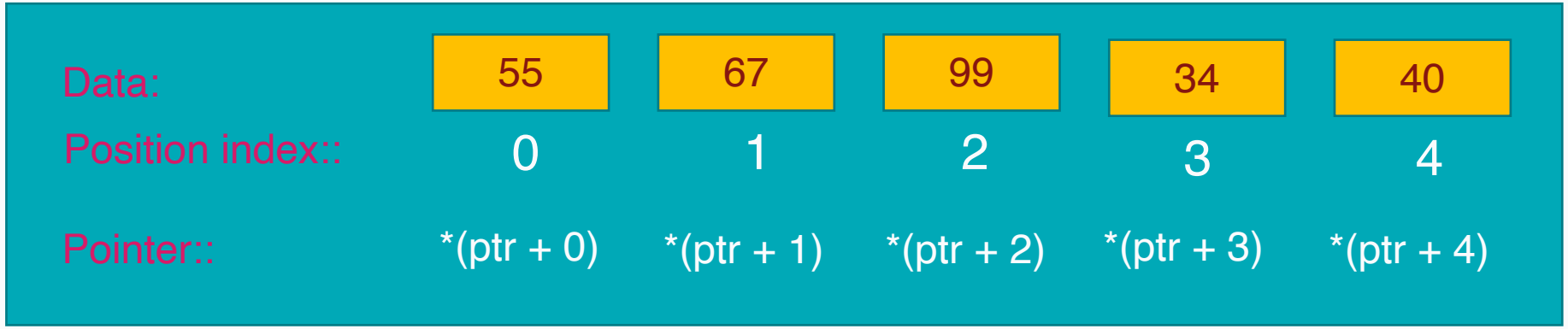
int project_grades [5]



Pointers and Arrays

int age [5]

int *prt = age;

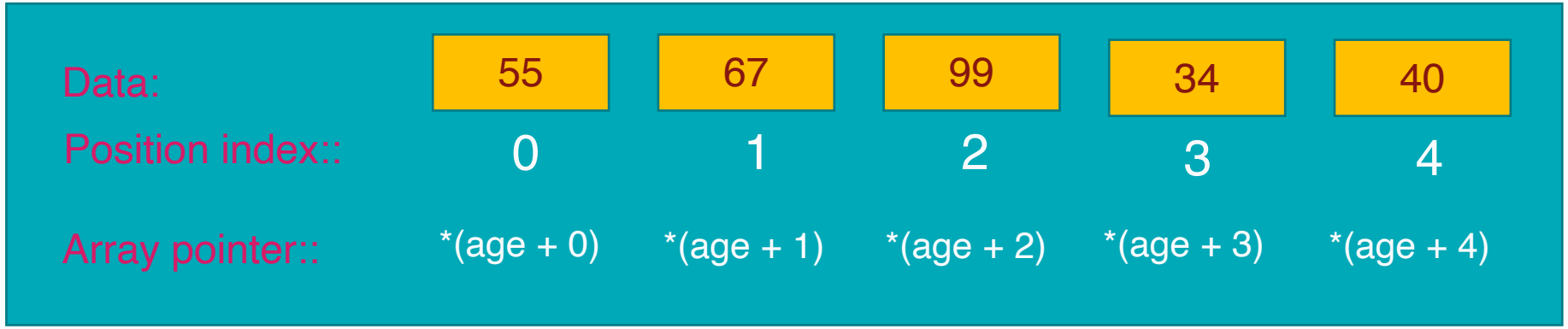


A pointer to an array can be incremented to point to the next element.

```
for (int i = 0; i < 5; i++) {
    printf("%d ", *(ptr + i));  // Access elements using pointer
}
```

Arrays using pointer notation

int age [5]



A pointer to an array can be incremented to point to the next element.

```c
// Accessing array using pointer notation
for (int i = 0; i < 5; i++)
{
    printf("%d ", *(age + i)); // Access elements using pointer
}
```

Program Example 1

- Create a program to fulfil the following requirements.
  - Ask a user to input the daily temperature recorded for the past 5 days (Monday to Friday).
  - Display the 5 temperatures on screen using Subscript notation
  - Display the 5 temperatures on screen using Pointer notation

Problem Solving

- Break the problem into smaller parts:
    1. Create the Array
    2. Populate the array with 5 numbers
    3. Print the array using a for loop (Subscript notation)
    4. Print the array using a for loop (Pointer notation)

Problem 1 Solution

problem_1.c > ...

```c
#include <stdio.h>

int main()
{
    int arr[5] = {1, 2, 3, 4, 5};
    int *ptr = arr; // 'ptr' points to the first element of 'arr'

    // Accessing array using subscript notation
    for (int i = 0; i < 5; i++)
    {
        printf("%d ", arr[i]); // Access elements using pointer
    }

    printf("\n-------------\n");

    // Accessing array using pointer notation
    for (int i = 0; i < 5; i++)
    {
        printf("%d ", *(ptr + i)); // Access elements using pointer
    }

    return 0;
}
```

- Break the problem into smaller parts:
  1. Create the Array
  2. Populate the array with 5 numbers
  3. Print the array using a for loop (Subscript notation)
  4. Print the array using a for loop (Pointer notation)

Mandatory Question Prog Lab 6 – Q4

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

4. Write a program to read in fifteen numbers and display them as follows:

(a) each number on a separate line
(b) on one line, each number separated by a single space
(c) as in (b) but in the reverse order to which they were input.

Problem Solving

- **Break the problem into smaller parts:**
  1. Create the Array
  2. Populate the array with 15 numbers
     - Use a for loop and scanf to get the values from the user
  3. Print each value on a separate line
     - For loop and printf, \n to put each on new line
  4. Print values on the same line
     - For loop and printf, space between each number
  5. Print values on the same line (in reverse)
     - For loop and printf, space between each number, starting at last position and decrementing downwards

Mandatory Question Solution

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TECHNOLOGICAL UNIVERSITY DUBLIN

mandatory.c > ...

```c
#include <stdio.h>

int main() {
    int numbers[15];  // Array to store the 15 numbers

    // Read 15 numbers from the user
    printf("Enter 15 numbers:\n");
    for (int i = 0; i < 15; i++) {
        scanf("%d", &numbers[i]);
    }

    // Display each number on a new line
    printf("\nEach number on a new line:\n");
    for (int i = 0; i < 15; i++) {
        printf("%d\n", numbers[i]);
    }

    // Display all numbers on one line, separated by a space
    printf("\nAll numbers on one line, separated by a space:\n");
    for (int i = 0; i < 15; i++) {
        printf("%d ", numbers[i]);
    }
    printf("\n");

    // Display all numbers in reverse order, separated by a space
    printf("\nAll numbers in reverse order, separated by a space:\n");
    for (int i = 14; i >= 0; i--) {
        printf("%d ", numbers[i]);
    }
    printf("\n");

    return 0;
}
```

- Can you make a suggested improvement for the above solution?

Programming Pitfall

To summarise the above, you now have **TWO** ways to access an array:

1. **Subscript notation**

   This is when you use [and] to access the elements of an array

   e.g., a[0], a[1], etc.,

2. **Pointer notation**

   This is when you use the dereference operator to access the array

   e.g., *a, *(a + 1), *(a + 2), etc.,

array_name[i] ≈ *(array_name + i)

Subscript notation Pointer notation

Questions