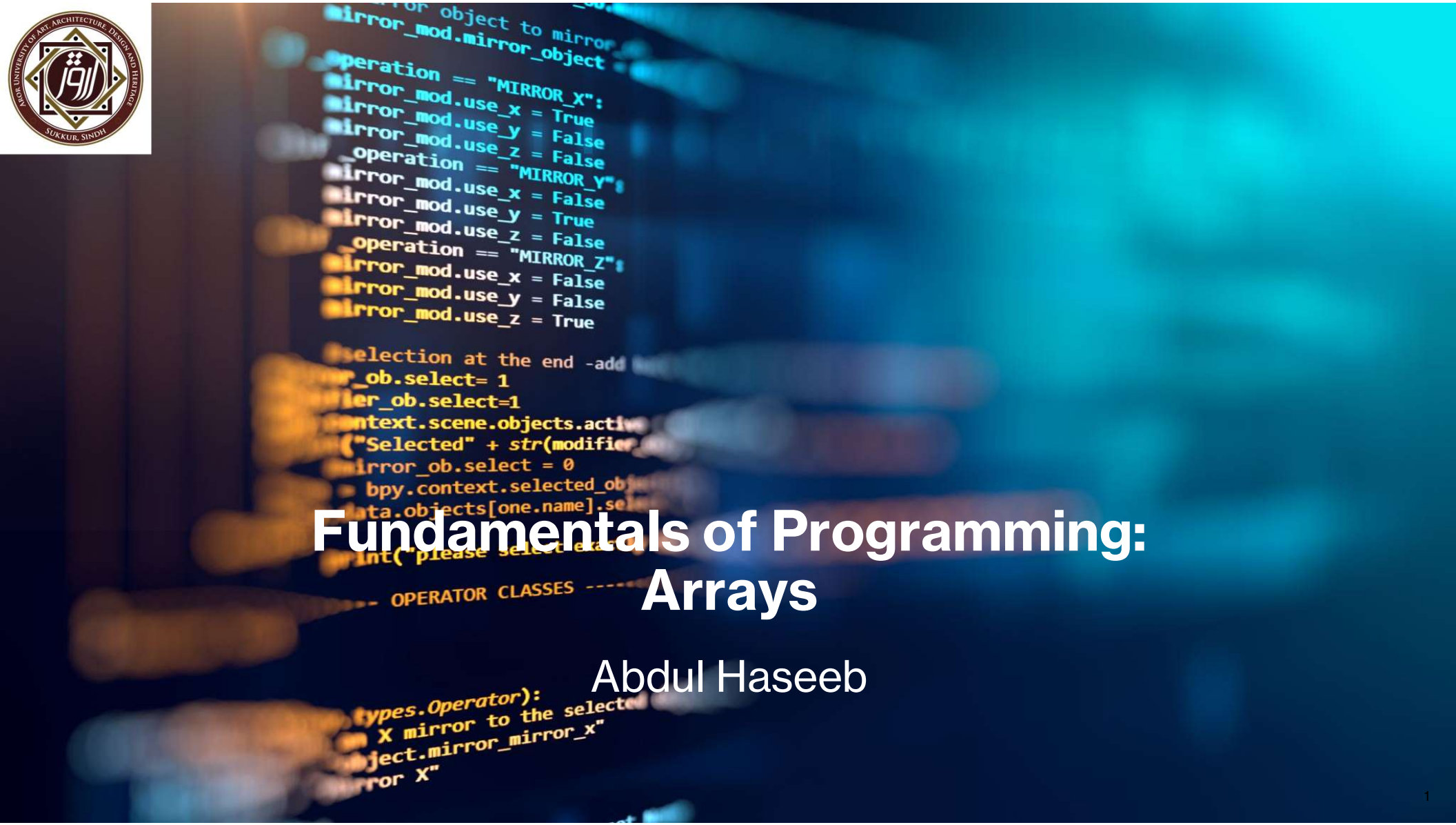


Fundamentals of Programming: Arrays

Abdul Haseeb

Agenda

- Why enumerations are used in C++
- What is enumeration
- Syntax for using enums
- Enumeration with Switch-case

Problem

Problem:

Store the price of 4 to 5 different products in a store.

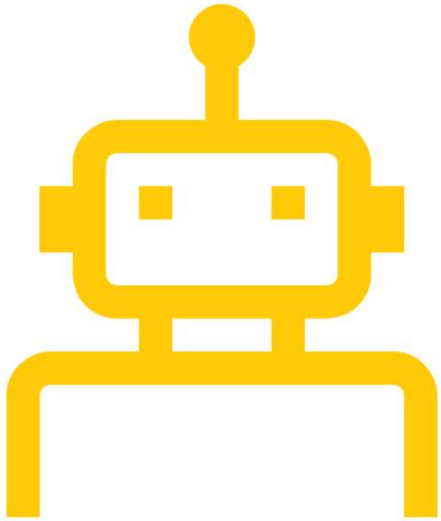
Solution:

Create 5 different variables:

- float price1;
- float price2;
- float price3;
- float price4;
- float price5;

Another Problem

- Store ages of all family members
- Let's say you have to store the names of students in a class
- Now let's expand it to the whole university



Solution

- Arrays
  - Avoid the repetitive task of declaring and initializing the variable

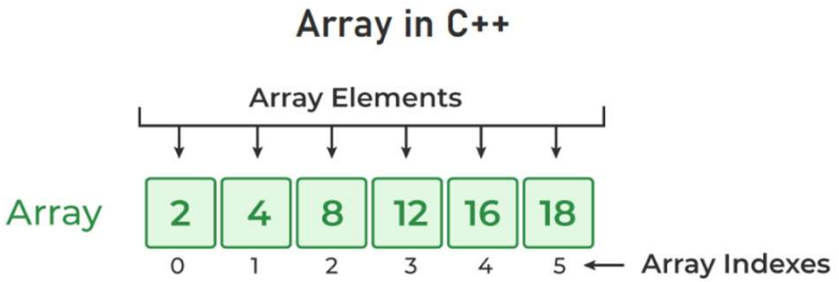
```
int ages[] = {71, 42, 37, 5, 18};
```

What is an Array?

- **Data Structure** that is used **to store multiple values** of **similar type** in a **contiguous memory** location
- An array is also called as homogenous collection of data
- **Examples:**
  - **Marks of students in a class**
  - **Collection of medicines in a medical store**

An array of 6 numbers

- Index of an array starts from 0 to n-1
- (where n is the size of an array)



Index

INDEX IS THE POSITION-1 WHERE AN ARRAY ELEMENT IS STORED

Declaration of an array

- int is the data type, you can use char, float, double etc
- arr is the name of array variable, which follows naming rules of a variable
- 5 inside the square brackets refers to the size of array, which can not be changed


```
data_type array_name[Size_of_array];
```

Example

```
int arr[5];
```

What do you guess how much bytes in memory an array will occupy?

- It depends on the data type and size of an array

Initialization of an array

MANY POSSIBLE WAYS LET'S DISCUSS SOME COMMON WAYS

1. Initialize Array with Values and size in C++


```
int arr[5] = {1, 2, 3, 4, 5};
```

1. Size of array is fixed in this case, can not exceed 5 elements

2. Initialize Array with Values and without Size in C++


```
int arr[] = {1, 2, 3, 4, 5};
```

1. Size of array is decided based on the number of elements inside curly braces

```
3
4  int main(){
5
6
7  int nums[5];
8
9  nums[0]=3;
10 nums[1]=4;
11 nums[2]=4;
12 nums[3]=4;
13 nums[4]=4;
```

3. Initialize with the help of indexes

Creating an array of names

Creating array of vowel letters

4. Initializing all the elements of an array with zero

- This method only works for zero

```
int zero_array[5] = {0};
```

Accessing an element of array

- Element can be accessed with the help of:
  - Array name
  - Along with an index number specified in the curly braces

Example: **cout<<nums[3];** //It will print 4th element of the nums array on console screen

Example of array element access

```cpp
//  C++ Program to Illustrate How to Access Array Elements
#include <iostream>
using namespace std;

int main()
{

    int arr[3];

    // Inserting elements in an array
    arr[0] = 10;
    arr[1] = 20;
    arr[2] = 30;

    // Accessing and printing elements of the array
    cout << "arr[0]: " << arr[0] << endl;
    cout << "arr[1]: " << arr[1] << endl;
    cout << "arr[2]: " << arr[2] << endl;

    return 0;
}
```

Traversing an array

- Traversing means to visit each element of array exactly once


```cpp
// C++ Program to Illustrate How to Traverse an A
#include <iostream>
using namespace std;

int main()
{

    // Initialize the array
    int table_of_two[10]
        = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };

    // Traverse the array using for loop
    for (int i = 0; i < 10; i++) {
        // Print the array elements using indexing
        cout << table_of_two[i] << " ";
    }

    return 0;
}
```

Exercise: Must complete before coming tomorrow

- Declare and Initialize an array of 6 integers, namely nums_array
- Print all the values of array without loop, with the help of array index
- Print all the values of array through for loop, with the help of array index