

Fundamentals of Programming: Arrays

Abdul Haseeb

```
for object to mirror_mod.mirror_object
    operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
= bpy.context.selected_objects
data.objects[one.name].select
print("please select object")

-- OPERATOR CLASSES -----

types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"
```

Agenda

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

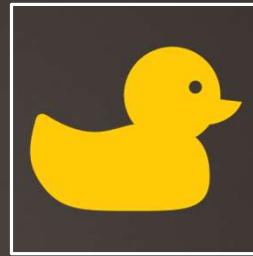
Problem

3



Problem:

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



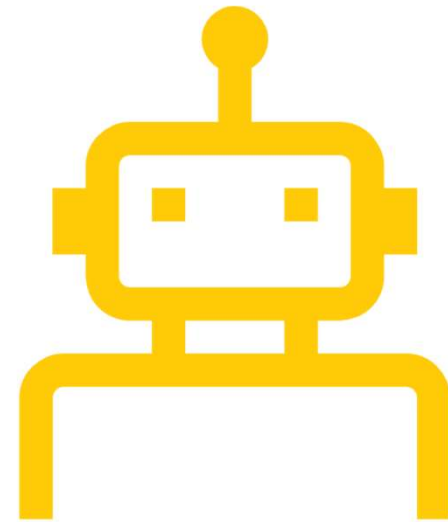
Solution:

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};
```

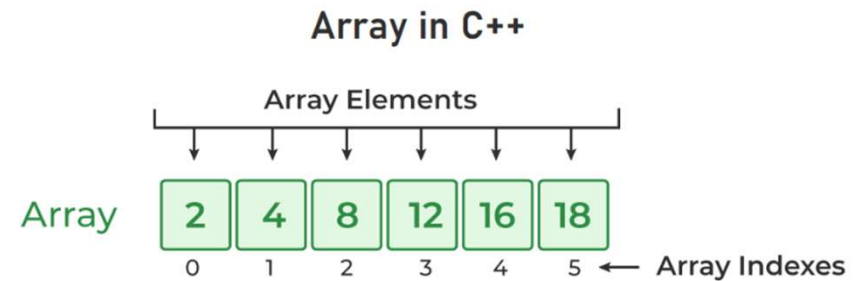
What is an Array?

- ▶ **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)

7



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



FQ J I X D Y M E B S L J B W X D U N L
G F B V W L C T F P O I Z Q A Y W H A T
M Y V L O Y F J R C V U N I J P N J H I
W Z U X Q U R A X I O M V M V O F T D C
V Y C D Y C J K M O P X E F R S P C O B
K B J I M U K I V A G V G R Q N T E Z X
Z H Y B S E C N I M D G O M F V E T O E
C I P U Y K F I X O C T F Z C H J E A R
Y K R V E C I O C R L X C L K L C T R D
Q L G Z R W F P F O E I Y F V R M Z H X
R P Z Y D U I V T E A X L J W S I R U C
J L A V M P L O T Y C K I B Q Y W Y P K
B P F R D J T V A Q I F S T Z V F M J C
S Y E C V I N G F B R N Y U C B S N T D
C F I B R M S Z J E D X R W T K A D F E

Creating
array of
vowel
letters

4. Initializing all the elements of an array with zero

17

- ▶ 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

```
// 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;
}
```

Example of array element access

Traversing an array

- ▶ Traversing means to visit each element of array exactly once

```
// C++ Program to Illustrate How to Traverse an Array
#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