



Fundamentals of Programming: Enumeration

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 the 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

Primitive vs User-defined data type

3

Aspect	Primitive Data Types	User-Defined Data Types
Definition	Pre-defined by the language	Created by the programmer
Size in memory	Compiler-dependent, fixed size	Size depends on the structure and members defined by the programmer
Initialization	Automatically initialized to a default value (e.g., 0 for integers)	No automatic initialization; values must be explicitly set by the programmer
Example	<code>`int`, `char`, `float`, `bool`</code>	<code>`class`, `struct`, `enum`, `union`</code> , custom data structures

Problem with primitive types

4

- ▶ There is no restriction on a value, like integer can take any value from integer type

Why Enumeration?

5

- ▶ Used when we want our variable to have one of the possible set of values
- ▶ Example: Coffee shop, only limited types, we can't restrict the values in primitive data types

What is enumeration?

- ▶ Enumeration or Enums is a user-defined data type:
 - ▶ Set of values are specified
 - ▶ Variable can take any one value from the specified values
 - ▶ Enum keyword is used to define enumeration

Syntax : enum enumerated-type-name{value 1, value 2, value 3value n };

Example : enum player { Ronaldo , Salah , Messi , Neymar };

Syntax

Syntax: `enum enumerated-type-name{value 1, value 2, value 3value n };`

Example: `enum player { Ronaldo , Salah , Messi , Neymar };`

By default, Ronaldo is 0 , Salah is 1 and so on. But we can change the default value as:

```
enum player { Ronaldo=4 , Salah=2 , Messi=8 , Neymar=1 };
```



Syntax


```
#include<iostream>
using namespace std;

enum months { January , February , March , April , May , June , July };
               0       1       2       3       4       5       6

int main()
{
    months m;

    m= March;

    cout<<"Month is : "<<m+1;

    return 0;
}
```

Output : 3

Example

Class activity

10



CREATE AN ENUMERATION
FOR WEEK DAYS



PRINT IT USING FOR LOOP

```
int main()
{
    // Defining enum Gender
    enum Gender { Male, Female };

    // Creating Gender type variable
    Gender gender = Male;

    switch (gender) {
    case Male:
        cout << "Gender is Male";
        break;
    case Female:
        cout << "Gender is Female";
        break;
    default:
        cout << "Value can be Male or Female";
    }
    return 0;
}
```

Example

Can we define enumeration inside any block or function

12

- ▶ Enumerations are named integer constants
- ▶ They are defined outside any function or block, at the global scope