Enumeration (or enum) in C

C enumeration (enum) is an enumerated data type that consists of a group of integral constants. **Enums** are useful when you want to assign user-defined names to integral constants. The **enum** keyword is used to define enums.

Defining and Declaring an Enum Type

Syntax

This is the syntax you would use to define an enum type –

```
enum enum_name{const1, const2, ... };
```

Enum Variable Declaration

Syntax

Below is the syntax to declare a variable of enum type –

```
enum enum_name var;
```

Example

Let us define an enum type with the name **myenum** –

```
enum myenum {val1, val2, val3, val4};
```

Identifier values are unsigned integers and they start from "0". val1 refers 0, val2 refers 1, and so on.

A variable of **myenum** type is declared as follows –

```
enum myenum var;
```

Change Enum Constants Values

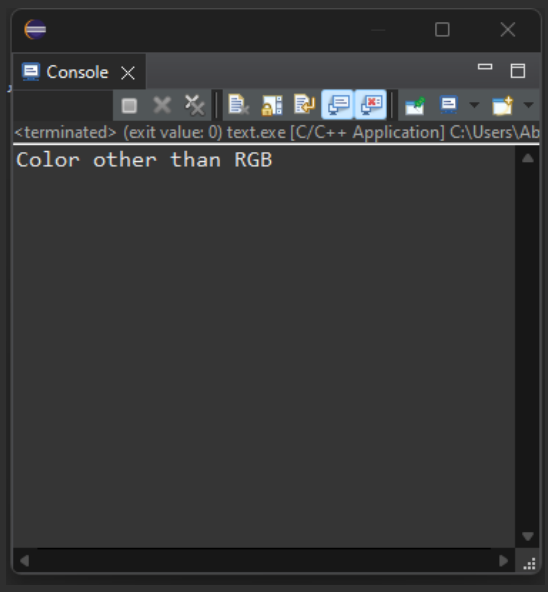
```c
#include <stdio.h>

enum status_codes { OKAY = 1, CANCEL = 0, ALERT = 2 };

int main() {
    // Printing values
    printf("OKAY = %d\n", OKAY);
    printf("CANCEL = %d\n", CANCEL);
    printf("ALERT = %d\n", ALERT);

    return 0;
}
```

```
OKAY = 1
CANCEL = 0
ALERT = 2
```

Enum in Switch Statements

C language switch case statement works with integral values. We can use enum type to define constants with (or, without) integral type values to use them in switch case statements.

```c
#include <stdio.h>

// Enum declaration
enum colors { VIOLET, INDIGO, BLUE, GREEN, YELLOW, ORANGE, RED };

int main() {
  // Enum variable declaration
  enum colors color = YELLOW;
  // switch statement using enum
  switch (color) {
    case BLUE:
      printf("Blue color");
      break;

    case GREEN:
      printf("Green color");
      break;

    case RED:
      printf("Red color");
      break;

    default:
      printf("Color other than RGB");
  }

  return 0;
}
```



Examples of Enumeration (enum) Type

Example 1: Enum Constants Get Incrementing Integer Values

main.c | text.c

```c
#include <stdio.h>

enum myenum {val1, val2, val3, val4};

int main(){

    enum myenum var;
    var = val2;
    printf("var = %d", var);

    return 0;
}
```

Console

```
<terminated> (exit value: 0) text.exe [C/C++ A
var = 1
```

Example 2: Enumerating the Weekdays

main.c | text.c

```c
#include <stdio.h>

int main(){

    enum weekdays {Sun, Mon, Tue, Wed,

    printf ("Monday = %d\n", Mon);
    printf ("Thursday = %d\n", Thu);
    printf ("Sunday = %d\n", Sun);
}
```

Console

```
<terminated> (exit value: 0) text.exe [C/C++ A
Monday = 1
Thursday = 4
Sunday = 0
```

Example 3: Declaring a Variable of Enum Type

```c
#include <stdio.h>

enum weekdays {Sun, Mon, Tue, Wed, Thu, Fri, Sat};

int main(){

    enum weekdays day;
    day = Wed;
    printf("Day number of Wed is = %d", day);

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ A
Day number of Wed is = 3
```

Example 4: Enum Values By Default Start from "0"

```c
#include <stdio.h>

enum months{Jan, Feb, Mar, Apr, May, Jun, Jul,
    Aug, Sep, Oct, Nov, Dec};

int main(){

    for (int i = Jan; i <= Dec; i++)
        printf("Month No: %d\n", i);

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ A
Month No: 0
Month No: 1
Month No: 2
Month No: 3
Month No: 4
Month No: 5
Month No: 6
Month No: 7
Month No: 8
Month No: 9
Month No: 10
Month No: 11
```

Example 5: Starting an Enum from 1

```c
#include <stdio.h>

enum months{Jan=1, Feb, Mar, Apr, May, Jun, Jul,
   Aug, Sep, Oct, Nov, Dec};

int main(){

   for (int i = Jan; i <= Dec; i++)
      printf("Month No: %d\n", i);

   return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ A
Month No: 1
Month No: 2
Month No: 3
Month No: 4
Month No: 5
Month No: 6
Month No: 7
Month No: 8
Month No: 9
Month No: 10
Month No: 11
Month No: 12
```

Example 6: Enumerating HTTP Status Codes

```c
#include <stdio.h>

enum status {
   OK = 200,
   BadRequest = 400,
   Unauthorized = 401,
   Forbidden = 403,
   NotFound = 404,
   InternalServerError = 500,
};

int main(){

   enum status code =  InternalServerError;
   if (code == 500)
   printf("Internal Server Error has been encountered");

   return 0;
}
```

```
Console
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\
Internal Server Error has been encountered
```

Example 7: Assigning a Fixed Number to Enum Constants

```c
#include <stdio.h>

enum myenum {a, b = 5, c, d, e = 10, f};

int main(){

    printf("a: %d\n", a);
    printf("b: %d\n", b);
    printf("c: %d\n", c);
    printf("d: %d\n", d);
    printf("e: %d\n", e);
    printf("f: %d\n", f);

    return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ App
a: 0
b: 5
c: 6
d: 7
e: 10
f: 11
```

Applications of Enums

- To store constant values, for example, weekdays, months, directions, colors, etc.
- Enums are used for using flags, status codes, etc.
- Enums can be used while using switch-case statements in C.