

# CSE 1320 - Intermediate Programming

## Enumerated Types

Alex Dillhoff

University of Texas at Arlington

# Enumerated Types

Enumerated types in C are scalar types (single values, as opposed to aggregate types).

These types are used often with structs and improve the readability and structure of a program.

They are analagous to created multiple constants with either `const` or the `define` preprocessor.

# Enumerated Types - Definition

In C, enumerated types are defined using the keyword `enum`.

```
enum boolean {  
    false,  
    true  
};
```

Unless specified, the values defined in an `enum` take on sequential values starting from 0.

# Enumerated Types - Variables

Once defined, variables can be created with enumerated types.

```
enum boolean is_prime = false;
```

A variable of an enumerated type should only be assigned valid values that belong to that type. However, compilers may not check that the assignment is valid.

# Enumerated Types - Manual Assignment

When defining enumerated types, you can specify scalar value for each constant.

```
enum http_status {  
    OK = 200,  
    MOVED_PERMANENTLY = 301,  
    BAD_REQUEST = 400,  
    FORBIDDEN = 403,  
    NOT_FOUND = 404  
};
```

# Enumerated Types - Usage

Once defined, enumerated types can be used with any operators that work with scalar values.

```
switch (result) {
    case OK:
        printf("Loading page.\n");
        break;
    case BAD_REQUEST:
        printf("Invalid request.\n");
        break;
    case NOT_FOUND:
        printf("Cannot find page.\n");
        break;
    default:
        break;
}
```

# Enumerated Types and `typedef`

There are two ways to `typedef` an enumerated type.

It can be done inline with the definition itself:

```
typedef enum {
    OK = 200,
    MOVED_PERMANENTLY = 301,
    BAD_REQUEST = 400,
    FORBIDDEN = 403,
    NOT_FOUND = 404
} http_status;
```

# Enumerated Types and `typedef`

It can also be done *after* the initial definition.

```
enum http_status {
    OK = 200,
    MOVED_PERMANENTLY = 301,
    BAD_REQUEST = 400,
    FORBIDDEN = 403,
    NOT_FOUND = 404
};

typedef enum http_status http_status;
```



# Enumerated Types and `typedef`

Note that, in the previous example, there are two different type names.

The first is `enum http_status` and the second is simply `http_status`.

You can use either to create a variable:

```
enum http_status status1 = OK;  
http_status status2 = OK;
```