Enumerated datatype:

Enumerated Datatype gives you an opportunity to

- Invent your own datatype
- What values can the variables of this datatype can take
- Allows numbers to be replaced by words

What is Enumerated Data Type???

Enumerated Data type gives you an opportunity to

- Invent your own data type
- Define what values of the variables of this data type can take
- The main purpose of he Enumerated data types is to allow numbers to be replaced by words

Enumerated type is a data type whose list of values is specified by the programmer.

SYNTAX

```
enum
enum_name

 {identifier_list}

enum_type;
```

EXAMPLE

```
enum
BOOLEAN
 {TRUE,FALSE
 }
b1,b2;
```

Another Way of having Syntax

SYNTAX

```
enum
enum_name

 {identifier_list}

enum_type;
```

EXAMPLE

```
enum
BOOLEAN
 {TRUE=0,
FALSE=1}
b1,b2;
```

Note:

If no integer values are specified then the left most word has integer value 0 and each one after that is incremented by one from that point. (0, 1, 2, 3, etc…) This also means that the left-most word is generally the smallest and the right-most word is generally the largest.

EXAMPLE

```
enum
BOOLEAN
{TRUE,FALSE
}
b1,b2;
```

- In the example the User defined data type BOOLEAN has been defined
- The new data type has two values. **TRUE** and **FALSE**
- The word **TRUE** has an Integer value 0
- The word **FALSE** has an Integer value 1

Another Example:

EXAMPLE

```
enum Weekdays
{ Monday = 1,
Tuesday,
Wednesday,
Thursday,
Friday
 };
```

EXAMPLE

```
enum Weekdays
{ Monday = 1,
Tuesday,
Wednesday=6,
Thursday,
Friday
 };
```

Guess the output:

```
Void main()
{
    enum WeekDays{Mon,tue,wed
    thurs,fri,sat,sun}days;
    int i;
    for(i=Mon;i<=fri;i++)
    {
        printf("\n %d",i);
    }
getch();
}
```

Output
0
1
2
3
4