3.5b ENUMERATION

Objectives

- To understand working with enum data type

Agenda

- Definition
- Syntax
- Examples

Enumeration

- User-defined data type consisting of integral constants
- Each integral constant is given a name
- Keyword *enum* is used to define enumerated data type

```
enum type_name{value1, ..., valueN};
```

type_name is the enumerated data type name or tag
value1, ..., valueN are values of type type_name

- By default, value1 will be equal to 0, value2 will be 1
  and so on.
  However, the programmer can change the default value

Declaration of enum Variable

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

enum boolean check;
```

- A variable check is declared which is of type enum boolean

Changing the default value of enum elements

```
enum suit{
        club = 0;
        diamonds = 10;
        hearts = 20;
        spades = 3;
        };
```

- The default values respectively for diamonds and hearts would be 1 and 2
- However, the default values can be changed as above

An Example

```
#include <stdio.h>
enum week{monday, tuesday, wednesday
        thursday, friday, saturday, sunday}
int main(void){
    enum week today;
    int day;
    for(day=monday; day<=sunday; day++)
        printf("\n%d", day);
    return 0;
};
```

- Output for the above code will be 0123456

An Example

```
#include <stdio.h>
enum allDays{Monday, Tuesday, Wednesday
            Thursday, Friday, Saturday, Sunday}
int main(void){
    enum allDays aDay;
    int i;
    printf(“Please enter day of the week (1-7)\n”);
    scanf(“%d”,&i);
    aDay = allDays(i-1);
    if(aDay=Sunday || aDay==Saturday)
        printf(“It is a weekend :) \n”);
    else
        printf(“It is not a weekend :( \n”);

    return 0;
};
```

Summary

- Discussed the syntax of enum with examples
- Discussed the how to use the enum datatype.