Day 1

Enum

- If we want improve readability of source code then we should use enum.
- To improve readability, we should give name to the constant/literal.
- If we want to give name to the constant then we should use enum.
- enum is keyword in C. It is introduced by ANSI in C89.

```
enum Color
{
    RED, GREEN, BLUE    //Enum Constant / Enumerator
    //RED = 0, GREEN = 1, BLUE = 2
};
```

- We can define multiple enums in a program but name of enum constant must be unique.
- How to create object?

```
enum Color clr;
```

typedef

- If we want to create alias for existing data type then we should use typedef.
- Using typedef, we can not define new data type rather we can give short name or meaningful name to the exisiting datatype.

```
#define TRUE    1
#define FALSE   0
typedef int BOOL;
```

- Alias for enum

```
enum Color
{
    RED, GREEN, BLUE
};
typedef enum Color Color_t;
enum Color color1;  //OK
Color_t color2; //OK
```

```
typedef enum Color
{
    RED, GREEN, BLUE
}Color_t;
Color_t color; //OK
```

- If we want to write menu driven program then we should use enum in C/C++.

Pointer Basics

- "&" is a unary operator which is used to get address of variable/object.
- If we want to store address then we should use pointer.
- Pointer is derived data type.
- "*" can be considered as unary as well as binary operator which is used to declare pointer.
- A variable, which stores address of another variable is called pointer.
- Pointer declaration :

```
int* ptr;   //OK
int * ptr;  //OK
int *ptr;   //OK : Recommended
```

- Intialization is the process of storing value inside variable during declaration.

```
int num1 = 10;  //Initialization
```

- Assignment is the process of storing value inside variable after its declaration.

```
int num1 = 10;  //Initialization
num1 = 20;      //Assignment
```

- Pointer initialization

```
int number = 10;            //Initialization
int *ptrNumber = &number;   //Pointer Initialization
```

- Pointer Assignment

```
int number = 10;    //Initialization
int *ptrNumber;     //Wild Pointer
ptrNumber = &number;    //Pointer Assignment
```

- Uninitialized pointer is called wild pointer.

```
#define NULL ( (void*)0)
```

- NULL is macro which is having 0 address.
- If we want to initialize pointer or to avoid dangling pointer then we should use NULL.

```
int *ptrNumber = NULL;
```

- If pointer contains NULL value then such pointer is called NULL pointer.
- Process of accesing value of the variable using pointer is called dereferencing.

```
int number = 10;    //Initialization
int *ptrNumber = &number;    //Initialization
printf("Number  :   %d\n", number); //10
printf("Number  :   %d\n", *ptrNumber); //10 : dereferencing
```

- Size of any type of pointer on:

1. 16 bit compiler : 2 bytes
2. 32 bit compiler : 4 bytes
3. 64 bit compiler : 8 bytes

- sizeof() is operator which is used to get size of object/variable.

Constant

- It is a keyword, introduced in C89.
- Once initialized, if we dont want to modify value of the variable then we should use const keyword.

```
const double PI = 3.14; //OK
const double INTERST_RATE;  //OK
```

- In C language, initializing constant variable is optional. We can modify value of such variable using pointer.
- We can not modify value of constant variable but we can read value of it.Hence constant variable is also called as read-only variable.