Type Conversion:

- When operands of different data types are used in an arithmetic expression, one of the operand data type is converted to the other type. That is known as type casting.
- Type casting is of two types:
  - **Implicit Conversion (automatic):** This conversion takes place automatically.
    The order of conversion is as follows:
    Char (lowest rank) → unsigned int → long int → unsigned long int →float→double→long double (highest rank).
    The final result of an expression is converted to the type of variable on the left of the assignment sign before assigning the value to it.
    Example:
    ```
    int k;
    float a;
    k=2/9;          (k=0)
    k=2.0/9;        (k=0)
    a=2/9;          (a=0.0)
    a=2.0/9;        (a=0.2222)
    ```
  - **Explicit Conversion:** Cast operation is used to overcome automatic conversion. The variable declared in specific data type converted into the required type.
    Syntax: (type-name) expression.
    Example:
    ```
    int m=10;
    float y;
    y=(float)m/4;  (y=2.5)
    ```

Comment Lines:

- Comment Lines enhances the readability and understandability of a program.
- Comment lines are not executed. The compiler ignores the comment lines.
- They also help the programmer to debug and test the program.
- A single line can be commented by using double forward slash (//) at the beginning of a line.
- **Example:** //This is function prototype.
- A multiple lines can be commented by using /* before the beginning of first line and */ after the end of last line.
- **Example:**
  ```
  /*This is function prototype.
  This function takes two integer arguments and returns first to the power second.*/
  ```

Control or decision making statements/Control Structure:

- A decision control instruction can be implemented in C by using the following types of statements:
  - if statement
  - switch statement
  - conditional operator statement
  - goto statement

Decision making with if statement:

- **Syntax1 (if):**
```
if (condition)
        statement(s);
```
- **Example:**
```
main()
{
        int num;
        printf("Enter a number greater than 10:");
        scanf("%d",&num);
        if(num>10)
                printf("\nYou have entered correct no.");
}
```
- **Output:**
```
Enter a number greater than 10:12
You have entered correct no.
```
- **Syntax2: (if else)**
```
if (condition)
        statement(s);
else
        statement(s);
```
- **Example:**
```
main()
{
        int num;
        printf("Enter a number:");
        scanf("%d",&num);
        if(num%2 = = 0)
                printf("\nNo is even");
        else
                printf("\nNo is odd");
}
```
- **Output:**
```
Enter a number:12
No is even.
```
- **Syntax3: (else if ladder)**
```
if (condition)
        statement(s);
else if (condition)
        statement(s);
else if (condition)
        statement(s);
. . . . . .
. . . . . .
else
        statement(s);
```
- **Example:** **Printing Grades for a given mark.**
```
main()
{
        int mark;
        printf("Enter a mark:");
        scanf("%d",&mark);
        if(mark>=90)
                printf("\nGrade=O");
        else if(mark>=80)
                printf("\nGrade=E");
```

```
        else if(mark>=70)
                printf("\nGrade=A");
        else if(mark>=60)
                printf("\nGrade=B");
        else if(mark>=50)
                printf("\nGrade=C");
        else if(mark>=40)
                printf("\nGrade=D");
        else
                printf("\nGrade=F");
}
```

- **Output:**

```
Enter a mark:82
Grade=E
```

- **Syntax4: (Nested if)**

```
if (condition1)
{
        if (condition2)
                statement(s);
        else
                statement(s);
else
{
        if (condition3)
                statement(s);
        else
                statement(s);
}
```

- **Example:** **Checking a given year is leap or not.**

```
main()
{
        int year;
        printf("Enter an year:");
        scanf("%d",&year);
        if(year%100 = =0)
        {
                if(year%400 = =0)
                        printf("\nLeap Year");
                else
                        printf("\nNot a Leap Year.");
        }
        else
        {
                if(year%4 = =0)
                        printf("\nLeap Year");
                else
                        printf("\nNot a Leap Year.");
        }
}
```

- **Output:**

```
Enter an year:1982
```

Not a Leap Year.

- The above program can be done by using logical operators in the following way.
- **Example:** **Checking a given year is leap or not.**

```
main()
{
        int year;
        printf("Enter an year:");
        scanf("%d",&year);
        if((year%100 = =0 && year%400 = =0) ||
                (year%100 ! =0 && year%4 = =0))
                printf("\nLeap Year");
        else
                        printf("\nNot a Leap Year.");

}
```

- **Output:** Enter an year:1982
  Not a Leap Year.