

User-Defined Data Type in C

A user-defined data type is one that is derived from an existing data type. The user-defined data types extend the primitive types that are already available in the programming language according to the [GATE 2023 syllabus for CSE](#).

User-defined data type in C is sometimes referred to as UDT. Users define the UDT based on their requirements. Let us see various types of user-defined data types in C.

Types of User-Defined Data Types in C

Having discussed the need and definition of user-defined data types in [C programming language](#), we can now individually understand all the available user-defined data types in C. The C programming language consists of the following types:

- Structures
- Union
- Typedef
- enum

Structure

The structure data type consists of related information regarding an object of a particular type. Every data item present in the structure is called a member. These are sometimes referred to as fields. The “struct” keyword is used for creating a structure and can be studied well in the [GATE notes](#).

The syntax of the structure is as follows:

```
struct name of structure type
{
data type1 variable_name2;
data type2 variable_name2;
};
```

Example:

```
struct person
{
string name;
```

```
int age;
```

```
};
```

Union

It is a collection of various different data types present in the C language that are not similar to each other. A union data type contains many members, but only one member can be accessed at a particular time. Among other user-defined data types in C mentioned in the [GATE exam syllabus](#), structure and union are very similar to each other. The keyword “union” is used to create a union data type.

The syntax of the union data type is as follows:

```
union name of the union type
```

```
{
```

```
data type1 variable_name1;
```

```
data_type2 variable_name2;
```

```
};
```

Example:

```
union person
```

```
{
```

```
string name;
```

```
int age;
```

```
};
```

Typedef

The keyword typedef is used in C programming to create an alternate or alias name for a data type already existing. It only refers to the existing data and does not create any new entries.

The **syntax** for typedef is as follows:

```
typedef existing_data_type alias_name;
```

Example:

```
void main()
```

```
{  
    typedef float decimal;  
    decimal weight= 58.9;  
}
```

Enum

It is referred to as an enumerated data type. A special data type consists of sets of values known as elements. Enum usually allows giving symbolic names to represent a list of constants that are related to each other.

The syntax of enum data type is as follows:

```
enum variable_name(value1, value2, value3,..... value n);
```

Example:

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

Need for User-Defined Data Types in C

The data types present in the program help us store data needed for the program execution. In C programming though we have primitive data types and derived data types, we can define the data type of our own, known as a user-defined data type in C.

For feeding the customized requirements of the user, user-defined data types are very useful and are important for the [GATE exam](#). Users can define data types according to their requirements. In scenarios where primitive and derived data types can not fulfill users' requirements, the user-defined data types come to the rescue.