Data Types

Data types in C specify the type of data that a variable can hold. C provides several basic data types, which can be broadly categorized into four types:

1. **Basic or Primitive Data Types** - **int:** Integer data type is used to store integer values.

```c
int myInt = 10;
```

- float: Float data type is used to store floating-point numbers.

```c
float myFloat = 3.14;
```

- double: Double data type is used to store double-precision floating-point numbers.

```c
double myDouble = 3.14;
```

```

- char:Char data type is used to store a single character.

```c
char myChar = 'A';
```

2.**Derived Data Types**:

- Array:An array is a collection of elements of the same data type.

```c
int myArray[5] = {1, 2, 3, 4, 5};
```

- Pointer: A pointer is a variable that stores the address of another variable.

```c
int *ptr;
int num = 10;
ptr = #
```
```

3. **Enumeration Data Type**:

- **enum:** Enumerations allow you to create a list of named integer constants.

```c
enum Days {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
enum Days today = Wed;
```

4. **User-defined Data Types**:

- struct: A structure is a user-defined data type that allows you to group variables of different data types.

```c
struct Point {
    int x;
    int y;
};
struct Point p1;
```

Constants in C:

Constants are fixed values that do not change during the execution of a program. In C, constants can be of two types:

1. **Numeric Constants**:

- Integer Constants: Integer constants can be written as a sequence of digits without a decimal point.

```c
const int myConstInt = 42;
```

- Floating-point Constants: Floating-point constants contain a decimal point or an exponent.

```c
const float myConstFloat = 3.14;
```

2. **Character Constants**:

- Character constants are enclosed in single quotes.

```c
const char myConstChar = 'A';
```

3. **String Constants**:

- String constants are sequences of characters enclosed in double quotes.

```c
const char myConstString[] = "Hello, World!";
```

4. **Symbolic Constants**:

- Symbolic constants are created using the `#define` preprocessor directive.

```c
#define PI 3.14
```

or using `const` keyword:

```c
const double PI = 3.14;
```

Using symbolic constants makes code more readable and maintainable

as it provides meaningful names to constants.

That's a brief overview of data types and constants in C. Remember to use appropriate data types based on the requirements of your program and to use constants for values that should not be changed during the execution of the program.