Datatypes in c:

1. char:
    a. Used to refer to characters or string
2. int:
    a. Used to refer to integers
3. float:
    a. Used to refer to decimal values, Precision is up to 6 digits after the comma.
4. double:
    a. Used to refer to decimal values, Precision is up to 15 digits after the comma.
5. void:
    a. Empty data or No data, it can be used in functions to specify return values.

Printing types, like printf("Test %_"); Where:

1. %d or %i = Decimal integer, like 1,2,3
2. %f = FLoating values, like 10.123
3. %c = character, Like A, b
4. %s = String, like "hello"
5. %p = to display the address of a variable.

When using **scanf** for user input, you need a: &variable, meaning that you need to specify the address of the variable that you want to store the user input at.
"Store the data given by the user to the address of the variable".

Operators:

- && = AND
- || = OR
- ! = NOT
- % = remainder

switch-case:

```
switch(expression) {
        case 1:
                printf();
                break;

        case2:
                printf();
                break;

        default:
                printf();
}
```

in a do-while loop:

- The code is executed no matter whether the condition is true or false, and then the condition of the while is checked. If condition is true, the keep going.

The difference is that in a while loop, the condition is checked first.

Arrays:

- int area[10]: This is an array containing 10 numbers (ints(
- int numbers[] = {1,2,3}: This given an array of length 3 containing these three elements.
- To access the array, you use area[0], so access first element.
- a 2-dimensional array: int matrix[2][2]: 2x2 matrix

Pointers:

- To get the memory address of a variable, we use the %variable_name, so use the %
- A variable that stores another variable's address is called a pointer
- A pointer has a star, meaning * before the variable.
- To store the address of any variable, write the variable name and then use & to assign the address to the variable, like this:
    - int like;
    - int *point;
    - point = &like;
- You can use a star before the pointer name to get the value at the address stored in the pointer variable. Star * is also known as the dereferencing operator.
- A pointer that stores the address of another pointer variable is called a double pointer.

String:

- A string is an array of characters

Structure:

- It's a collection of different data items:

Union:

- This is a special data type in C that allows to store different data types in the same memory location.
- Instead of struct keyword, we have to use the union keyword. But in union, you can only have one union member with the value at a time. After you assign the value to another union member, then the value of the previous union member is deleted.
- Union is used for efficient memory management

Pointer:

- a pointer is a "variable like" reference that holds a memory address to another variable, array etc.
- Ex:
    - int *pAge = &age
    - This means that the pointer has its own address, but the value stored within it is an address.
    - And we can access the value at this address by using the indirection operator (value at address).

- To dereference a pointer you need to type the pointer name and with the pointer * before the name.
- So for example: printf(“Value stored at address: %d”, *pAge)
- By dereferencing the pointer in the printf above, we’re extracting the value at this given address *pAge.
- So you use the * operator when you want to declare a pointer as well as when you want to access a value at the stored address.