Structs

1. **What is a struct?**
    - A struct in C is like a container that holds multiple related pieces of information together.
    - It allows you to create a custom data type by combining different variables under one name.
2. **How to define a struct?**
    - To create a struct, you use the `struct` keyword followed by the struct name.
    - Inside the struct, you list the variables and their data types.
    - For example:

```
struct Person {
    char name[50];
    int age;
    double height;
};
```

3. **Creating variables of a struct type:**
    - Once you define a struct, you can create variables of that struct type, just like creating variables of basic data types but you need to write the keyword `struct` first.
    - For example:

```
struct Person p; // Creating a variable named "p1" of type "Person"
```

4. **Accessing struct members:**
    - You can access the members of a struct using the dot `.` operator.
    - For example:

```
strcpy(p.name, "Mohamed"); // Assigning a value to the "name" member
p.age = 25; // Assigning a value to the "age" member
p.height = 1.75; // Assigning a value to the "height" member
```

5. **Passing structs to functions:**

   - You can pass struct variables as arguments to functions, allowing you to work with the entire struct data within the function.
   - For example:

```
void printInfo(struct Person p) {
    printf("Name: %s\n", p.name);
    printf("Age: %d\n", p.age);
    printf("Height: %.2f\n", p.height);
}
```

Notes:

- Structs don’t allow you to put a function inside them, and all members are public.
- To use `strcpy()` you have to include `string.h` header file.