Procedural Programming vs. Object-Oriented Programming (OOP) Paradigm:

Procedural Programming:

- Focuses on procedures or functions.
- Data and functions are separate entities.
- Emphasizes top-down design.
- Typically uses functions to manipulate data.
- C and Pascal are examples of languages that primarily use procedural programming.

Object-Oriented Programming (OOP):

- Focuses on objects, which are instances of classes.
- Data and functions (methods) are encapsulated within classes.
- Emphasizes modularity and reusability through classes and objects.
- Promotes concepts like inheritance, polymorphism, and encapsulation.
- C++, Java, and Python are examples of languages that support OOP.

Structures vs. Classes in C++:

Structures:

- Used for grouping related data members.
- Members are public by default.
- Typically used for simple data structures.
- Cannot have member functions (methods).

```
struct Point {

   int x;

   int y;

};
```

Classes:

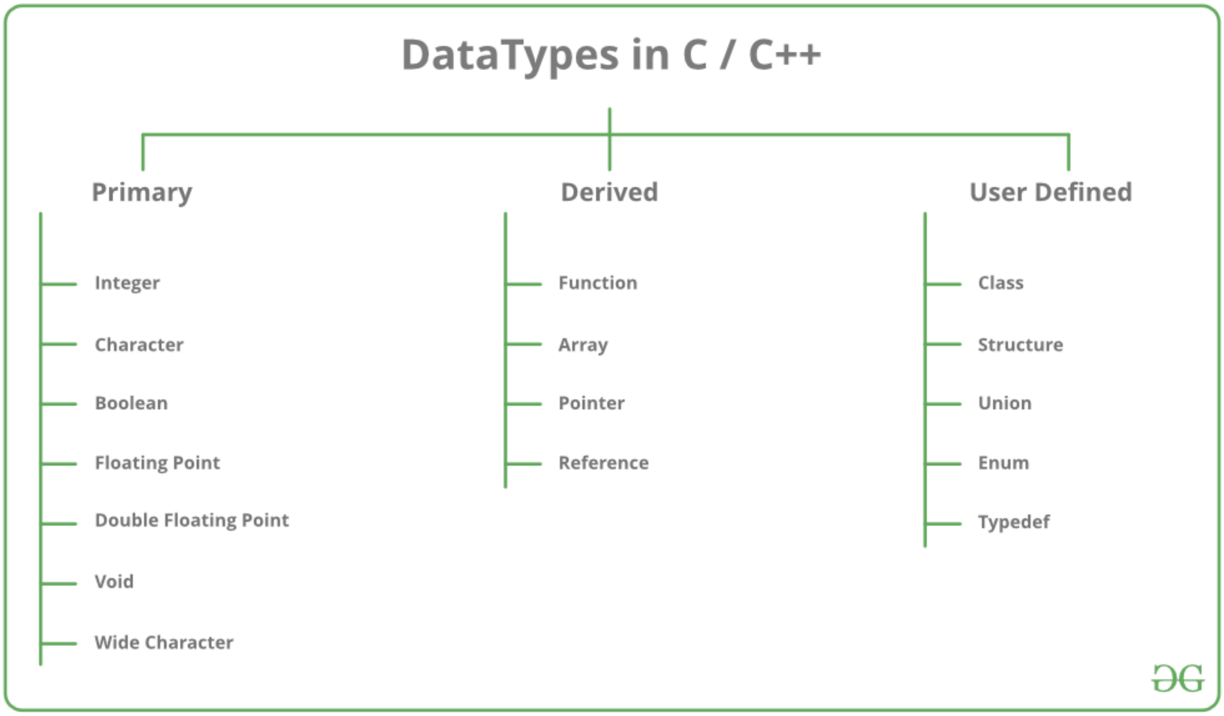
- Used for grouping data members and member functions.
- Members are private by default (encapsulation).
- Used for creating complex data structures with behavior.

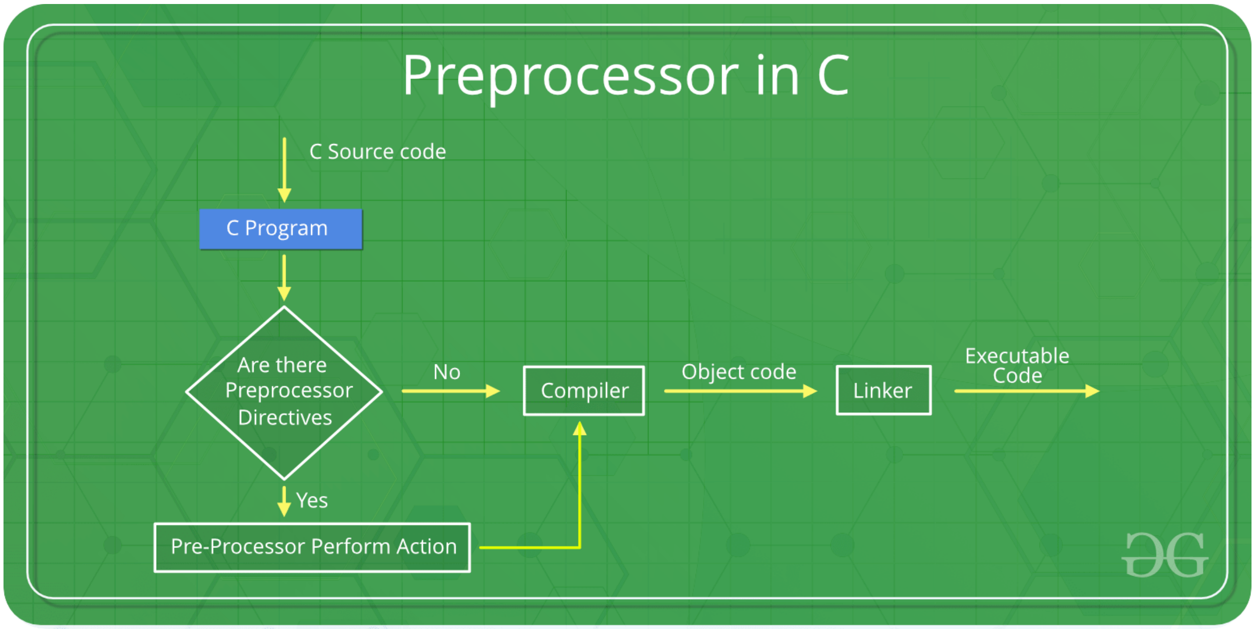
- Can have member functions (methods).

```
class Circle {
private:
    double radius;

public:
    Circle(double r) : radius(r) {}
    double area() {
        return 3.14 * radius * radius;
    }
};
```

Data Types In c++:-





There are 4 Main Types of Preprocessor Directives:

1. Macros **`#define`** *`token value`*
2. File Inclusion **`#include`** *`<file_name>`*
3. Conditional Compilation
4. Other directives