

COVID-19 Management System

Made By-

Karan Malhotra(CO19333)

Jagteshvar Jot Singh(CO19330)

Pryant Pandit(CO19348)

Concepts Used in the Project

Its usage in the program is illustrated -
`int main()`

1)File Handling -

Files are used to store data in a storage device permanently. File handling provides a mechanism to store the output of a program in a file and to perform various operations on it.

```
input
input
display //displaying
fstream
```

Its usage in the program is illustrated below-

2)Structure

A structure is a user-defined data type in C/C++. A structure creates a data type that can be used to group items of possibly different types into a single type.

```
struct
char      30
int
char      7
char      20
char      200
char      10
```

3)Operator Overloading -

In C++, we can make operators to work for user defined classes. This means C++ has the ability to provide the operators with a special meaning for a data type, this ability is known as operator overloading.

Its usage is illustrated below-

```
String operator String //overloading
```

```
String  
strcat str str  
strcpy str str  
return
```

4)String Header File

C++ has in its definition a way to represent **sequence of characters as an object of class**. This class is called std::string. String class stores the characters as a sequence of bytes with a functionality of allowing **access to single byte character**.

Its usage in the program is illustrated below-

char

char
strcpy str

5)Constructor Overloading

In C++, We can have more than one constructor in a class with same name, as long as each has a different list of arguments. This concept is known as Constructor Overloading and is quite similar to [function overloading](#).

Its usage in the program is illustrated below-

char

strcpy char str

6)Classes-

A **class** in **C++** is a [user-defined type](#) or [data structure](#) declared with [keyword](#) `class` that has data and functions (also called [member variables](#) and [member functions](#)) as its members whose access is governed by the three [access specifiers](#) *private*, *protected* or *public*. By default access to members of a C++ class is *private*. The private members are not accessible outside the class; they can be accessed only through methods of the class. The public members form an [interface](#) to the class and are accessible outside the class.

Its usage in the program is illustrated below-

class

```
char    20    //member variable for
string input
public
void
```

```
cout    "enter string"
cin     str 20
```