

The background features a gradient from dark purple at the top to dark blue at the bottom. Overlaid on this are several faint, light-colored circular patterns. On the left side, there is a large circular scale with tick marks and numbers ranging from 140 to 260. Other smaller circular elements with arrows and partial scales are scattered across the background.

OBJECT-ORIENTED PROGRAMMING (OOP)

KIAN _ ACADEMY

- ▶ C++ is an object-oriented programming language
- ▶ Everything in C++ is associated with classes and objects, with its attributes and methods

Example : in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake .

- ▶ Attributes and methods are basically variables and functions that belongs to the class
- These are often referred to as "class members".

Note :

- ▶ The class members are data and operation
- ▶ A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects

CREATE A CLASS

▶ To create a class, use the class keyword

Example : Create a class called "MyClass":

```
class MyClass {    // The class
    public:        // Access specifier
        int myNum;    // Attribute (int variable)
        string myString; // Attribute (string variable)
};
```

Example explained :

The **class keyword** is used to create a class called MyClass.

The public keyword is an access specifier, which specifies that members (attributes and methods) of the class are accessible from outside the class. You will learn more about access specifiers later

Inside the class, there is an integer variable myNum and a string variable myString. When variables are **declared within a class**, they are called **attributes**

At last, end the class definition with a **semicolon ;**

Multiple Objects-:

You can create multiple objects of one class:

Example

```
// Create a Car class with some attributes
```

CREATE AN OBJECT

- ▶ In C++, an object is created from a class. We have already created the class named MyClass, so now we can use this to create objects
- ▶ To create an object of MyClass, specify the class name, followed by the object name
To access the class attributes (myNum and myString), use the dot syntax (.) on the object:

Example : Create an object called "myObj" and access the attributes:

```
#include <iostream>
#include <string>
using namespace std;
class MyClass {    // The class
public:           // Access specifier
    int myNum;    // Attribute (int variable)
    string myString; // Attribute (string variable)
};
```

```
int main() {
    MyClass myObj;    // Create an object of MyClass
    // Access attributes and set values
    myObj.myNum = 15;
    myObj.myString = "Some text";
    // Print values
    cout << myObj.myNum << "\n";
    cout << myObj.myString;
    return 0;
}
```

output:

15

Some text

Multiple Objects-:

You can create multiple objects of one class:

Example : Create a Car class with some attributes

```
class Car {  
    public:  
        string brand;  
        string model;  
        int year;  
};  
  
int main() {  
    // Create an object of Car  
    Car carObj1;  
    carObj1.brand = "BMW";  
    carObj1.model = "X5";  
    carObj1.year = 1999;
```

```
// Create another object of Car
```

```
Car carObj2;  
carObj2.brand = "Ford";  
carObj2.model = "Mustang";  
carObj2.year = 1969;
```

```
// Print attribute values
```

```
cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year  
    << "\n";  
cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year  
    << "\n";  
return 0;  
}
```

Output:

1_BMW X5 1999

2_Ford Mustang 1969

EXERCISES:

1 / Create an object of MyClass called myObj

```
..... ;
```

2 / Use an access specifier to make members of MyClass accessible from outside the class.

```
class MyClass {  
    ----- :  
    int myNum;  
};
```

3 / Create an object of MyClass called myObj, and use it to set the value of myNum to 15.

```
class MyClass {  
    public:  
        int myNum;  
};  
  
int main() {  
    1_-----  
    2_----- . -----  
    cout << myObj.myNum;  
    /*The Output is....?  
    return 0;  
}
```

4 / Create a function named myMethod with no return value. Then Use myObj to call myMethod inside main.

```
class MyClass {  
public:  
    -----() {  
        cout << "Hello World!";  
    }  
};  
  
int main() {  
    MyClass myObj;  
    -----();  
    return 0;  
}
```

Notes:

- 1 / An Object is an instance of a Class.
- 2 / When a class is defined, no memory is allocated but when it is instantiated (object is created) memory is allocated

KIAN_ACADEMY