1. CLASS AND OBJECT

AIM:

To create a class program and object in c++.

ALGORITHM:

· Include the header file <iostream> for input and output operations.

· Use the standard namespace std.

· Define a class named Student.

· Declare age as an integer data member inside the class.

· Declare name as a string data member inside the class.

· Begin the main() function.

· Declare an object S1 of the class Student.

· Assign the value 20 to the data member age of object S1.

· Assign the value "MUTHUGANESH S" to the data member name of object S1.

· Display the value of age using the output statement.

· Display the value of name using the output statement.

PROGRAM:

```
/*
  Program to demonstrate a simple class in C++
  Author   : MUTHUGANESH S
  Date     : 21/1/2026
  Filename: ClassProgram.cpp
  retval   : void
*/

#include <iostream>
using namespace std;

// Class Declaration
class Student{
    public:
        //Members Variables
        int age;
        string name;
};

// Main Function
int main(void) {
```

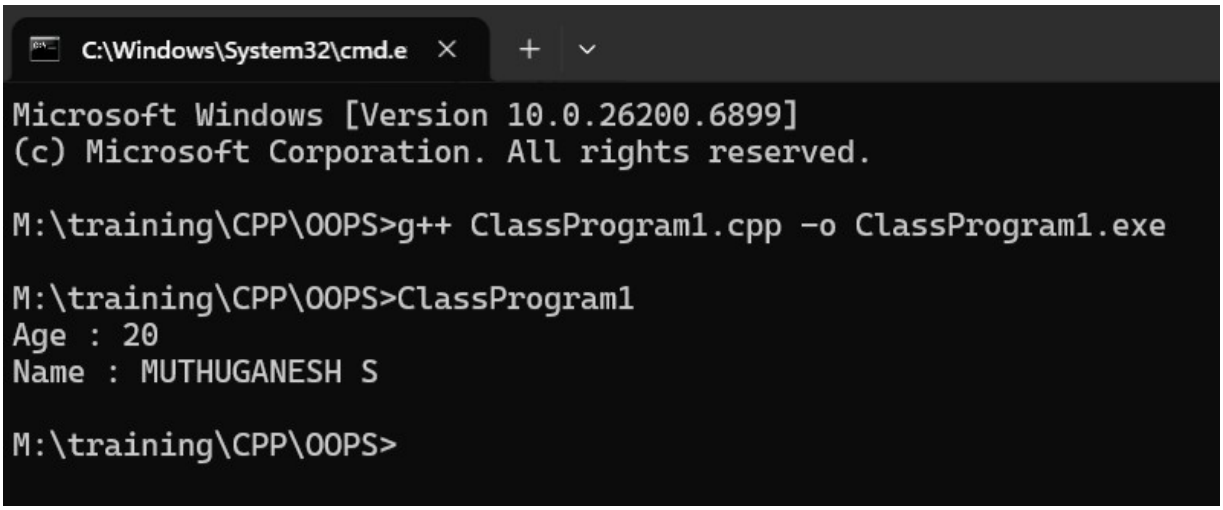
```
    // Object Declaration
    Student S1;

    // Assigning values to member variables
    S1.age = 20;
    S1.name = "MUTHUGANESH S";

    // Displaying values
    cout << "Age : "<< S1.age <<"\nName : "<< S1.name << endl;

    return 0;
}
```

OUTPUT:



2. ACCESS SPECIFIER

AIM:

To create Class and object in C++ program using Access specifier.

ALGORITHM:

· Include the header file <iostream> for input and output operations.

· Use the standard namespace std.

· Define a class named Student.

· Declare private data members Age and Name inside the class.

· Declare the member function setDetails() to assign values to Age and Name.

· Declare the member function display() to print the values of Age and Name.

· Begin the main() function.

· Declare an object S1 of the class Student.

· Call the member function setDetails() using object S1 to assign values.

· Call the member function display() using object S1 to display the values.

PROGRAM:

```
/*
*  Program to demonstrate the use of classes and Pointer to Object in C++
*  Author   : MUTHUGANESH S
*  Date     : 21/1/2026
*  Filename: ClassProgram3.cpp
*  retval   : void
*/

#include <iostream>
using namespace std;

// Class Declaration
class Student {
    public:
        int age;
        string name;
        void setDetails(int age, string name) {
            this->age = age;
            this->name = name;
        }
        void display() {
            cout << "Age : "<< age <<"\nName : "<< name << endl;
```

```
        }
};

// Main Function
int main(void) {

    // Object Declaration
    Student S1;

    // Pointer to Object
    Student *ptr;

    // Assigning address of S1 to ptr
    ptr = &S1;

    // Using pointer to set values
    ptr->setDetails(20, "MUTHUGANESH S");

    // Using pointer to display values
    cout<< "Using Pointer to Object:\n";
    ptr->display();

    cout<< "\nUsing Object Directly:\n";
    S1.display(); // Directly using object to display values

    return 0;
}
```

OUTPUT:

```
M:\training\CPP\OOPS>g++ ClassProgram2.cpp -o ClassProgram2.exe

M:\training\CPP\OOPS>ClassProgram2
Age : 20
Name : MUTHUGANESH S

M:\training\CPP\OOPS>
```

3. OBJECT REFERENCE

AIM:

To create a Class and Object and also Object pointer.

ALGORITHM:

· Include the header file <iostream> for input and output operations.

· Use the standard namespace std.

· Define a class named Student.

· Declare public data members age and name inside the class.

· Define the member function setDetails() to assign values using the this pointer.

· Define the member function display() to print the values of age and name.

· Begin the main() function.

· Declare an object S1 of the class Student.

· Declare a pointer ptr to the class Student.

· Assign the address of object S1 to the pointer ptr.

· Call the member function setDetails() using the pointer ptr.

· Display a message indicating the use of pointer to object.

· Call the member function display() using the pointer ptr.

· Display a message indicating direct object access.

· Call the member function display() using the object S1.

· Return 0 to terminate the program.

PROGRAM:

```
/*
*  Program to demonstrate the use of classes and Pointer to Object in C++
*  Author  : MUTHUGANESH S
*  Date    : 21/1/2026
*  Filename: ClassProgram3.cpp
*  retval  : void
*/

#include <iostream>
using namespace std;

// Class Declaration
```

```cpp
class Student {
    public:
        int age;
        string name;
        void setDetails(int age, string name) {
            this->age = age;
            this->name = name;
        }
        void display() {
            cout << "Age : "<< age <<"\nName : "<< name << endl;
        }
};

// Main Function
int main(void) {

    // Object Declaration
    Student S1;

    // Pointer to Object
    Student *ptr;

    // Assigning address of S1 to ptr
    ptr = &S1;

    // Using pointer to set values
    ptr->setDetails(20, "MUTHUGANESH S");

    // Using pointer to display values
    cout<< "Using Pointer to Object:\n";
    ptr->display();

    cout<< "\nUsing Object Directly:\n";
    S1.display(); // Directly using object to display values

    return 0;
}
```

OUTPUT:

```
M:\training\CPP\OOPS>g++ ClassProgram3.cpp -o ClassProgram3.exe

M:\training\CPP\OOPS>ClassProgram3
Using Pointer to Object:
Age : 20
Name : MUTHUGANESH S

Using Object Directly:
Age : 20
Name : MUTHUGANESH S
```

4. MULTIPLE OBJECT

AIM:

To create a class and multiple objects for same class.

ALGORITHM:

· Include the header file <iostream> for input and output operations.

· Use the standard namespace std.

· Define a class named Student.

· Declare public data members age and name inside the class.

· Define the member function setDetails() to assign values to age and name.

· Define the member function display() to print the values of age and name.

· Begin the main() function.

· Declare three objects S1, S2, and S3 of the class Student.

· Call the member function setDetails() for object S1 to assign values.

· Call the member function setDetails() for object S2 to assign values.

· Call the member function setDetails() for object S3 to assign values.

· Display a heading for Student 1 details.

· Call the member function display() for object S1.

· Display a heading for Student 2 details.

· Call the member function display() for object S2.

· Display a heading for Student 3 details.

· Call the member function display() for object S3.

PROGRAM:

```
/*
*  Program to demonstrate the use of classes and multiple objects in C++
*  Author   : MUTHUGANESH S
*  Date     : 21/1/2026
*  Filename: ClassProgram4.cpp
*  retval   : void
*/

#include <iostream>
using namespace std;
```

```cpp
// Class Declaration
class Student {
    public:
        int age;
        string name;
        void setDetails(int age, string name) {
            this->age = age;
            this->name = name;
        }
        void display() {
            cout << "Age : "<< age <<"\nName : "<< name << endl;
        }
};

int main(void) {
    // Object Declaration
    Student S1, S2, S3;

    // Assigning values to member variables
    S1.setDetails(20, "MUTHUGANESH S");
    S2.setDetails(21, "Magesh K");
    S3.setDetails(22, "Yogesh L");

    // Displaying values
    cout<< "Details of Student 1:\n";
    S1.display();
    cout<< "Details of Student 2:\n";
    S2.display();
    cout<< "Details of Student 3:\n";
    S3.display();

    return 0;
}
```

OUTPUT:

```
M:\training\CPP\OOPS>g++ ClassProgram4.cpp -o ClassProgram4.exe

M:\training\CPP\OOPS>ClassProgram4
Details of Student 1:
Age : 20
Name : MUTHUGANESH S
Details of Student 2:
Age : 21
Name : Magesh K
Details of Student 3:
Age : 22
Name : Yogesh L

M:\training\CPP\OOPS>
```