|  |  |
| --- | --- |
| **Ex. No. : 1** | **Classes and Objects** |
| **Date :** |  |
| **Aim:** |  |
| To write a C++ program to show how to create a class and how to create objects. | |
| **Theoretical Concepts:** |  |
| **Classes** |  |
| A class is a blueprint that defines the variables and the methods common to all objects of a certain kind. A class describes both the properties (data) and behaviors (functions) of objects. Classes are not objects, but they are used to instantiate objects.  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.  A class is defined in C++ using keyword ***class*** followed by the name of class. The body of class is defined inside the curly brackets and terminated by a semicolon at the end as shown in the syntax.  **Syntax:**  class ClassName  {  // some data  // some functions  }; | |
| **Objects** |  |
| In object-oriented programming languages like C++, the data and functions (procedures to manipulate the data) are bundled together as a self-contained unit called an object. Instances of a class data type are known as objects. All the public members of the class can be accessed through object. Object is a runtime entity, it is created at runtime. In other words, object is an entity that has state and behavior. Here, state means data and behavior means  functionality. | |

|  |
| --- |
| When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. To use the data and access functions defined in the class, you need to create objects using the following syntax.  **Syntax**  ClassName ObjectName-1, ObjectName-2, …, ObjectName-N; |
| **Accessing Data Members and Member Functions** |
| The data members and member functions of class can be accessed using the dot(„.‟) operator with the object.  **Syntax**  ObjectName.FunctionName(Parameter);  For example if the name of object is obj and you want to access the member function with the name printName() then you will have to write obj.printName()  The public data members are also accessed in the same way given however the private data members are not allowed to be accessed directly by the object. Accessing a data member depends solely on the access control of that data member. In C++, this access control is given by access specifiers **private**, **protected** or **public**.  **Syntax**  ObjectName.PublicMember  For example if the name of object is obj and you want to access the data member with the name totalmarks then you will have to write obj.totalmarks to access the content. |
| **Program:** |
| #include <iostream.h>  #include <conio.h>  class Distance  {  private:  int feet;  float inches; |

|  |
| --- |
| public:  void setdist(int ft, float in)  {  feet = ft; inches = in;  }  void getdist()  {  cout << "\nEnter feet: "; cin >> feet;  cout << "Enter inches: "; cin >> inches;  }  void showdist()  {  cout << feet << "\'-" << inches << "\" ";  }  };  int main()  {  Distance dist1, dist2; clrscr(); dist1.setdist(11, 6.25); dist2.getdist(); clrscr();  cout << "\ndist1 = "; dist1.showdist(); cout << "\ndist2 = "; dist2.showdist(); getch();  return 0;  } |
| **Sample Input and Output:** |
| Enter feet: 10  Enter inches: 4.75  dist1 = 11'-6.25"  dist2 = 10'-4.75" |
| **Result:** |
| Thus, a C++ program has been written to show how to create a class and how to create objects. |