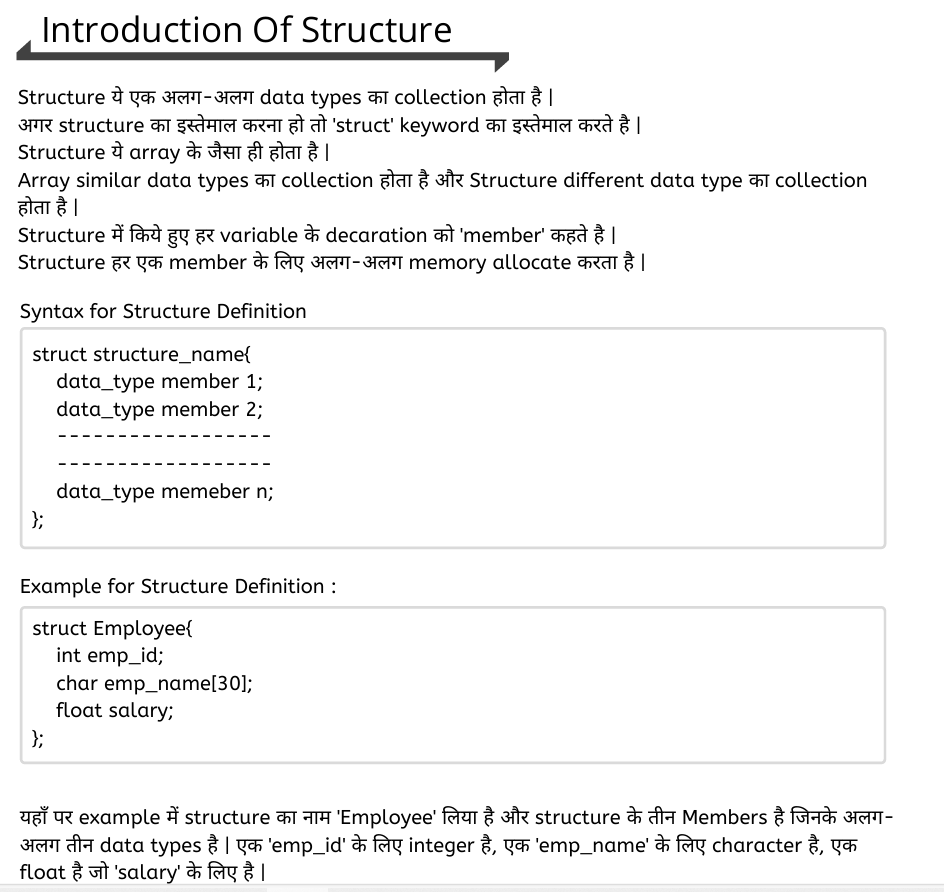
C++ OPPs



|  |
| --- |
| #include<iostream>  using namespace std;  // Define a struct named 'pop'  struct pop {      int a = 100; // Initialize 'a' to 100  };  // Define a class named 'top'  class top {  public:      int a = 10; // Initialize 'a' to 10  };  int main() {      // Creating an object 't' of struct 'pop'      pop t;      cout << t.a << "\n"; // Output: 100      // Creating an object 'p' of class 'top'      top p;      cout << p.a; // Output: 10      return 0; // Return statement at the end of main function  } |
| Output  100  10 |

|  |  |
| --- | --- |
| #include<iostream>  using namespace std;  class bhopal{      public:void show()  //public is acces specifier      {          cout<<"Welcome"; //it will take memory on runtime      }  };  int main(){      bhopal b;  //object call      b.show();   //object ke throw function call    }   |  | | --- | |  | |

|  |  |
| --- | --- |
| #include<iostream>  using namespace std;  class bhopal{      int a;      public:void show()  //public is acces specifier      {          a++;          cout<<"Welcome"; //it will take memory on runtime      }  };  int main(){      bhopal b;  //object call      b.show();   //object ke throw function call      cout<<sizeof(b);  }   |  | | --- | | Output  Welcome4 | |
| #include<iostream>  using namespace std;  class{      public:void show()  //public is acces specifier      {          cout<<"Welcome"; //it will take memory on runtime      }  }j; //anonymous function  int main(){      j.show();  }  Output  Welcome |
| #include <iostream>  using namespace std;  class top1  {  public:      void show(int n) // public is acces specifier      {          for (int i = 1; i <= 10; i++)          {              cout << i \* n << "\n";          }      }  };  int main()  {      int n;      cout << "Enter number ";      cin >> n;      top1 t1;      t1.show(n);  }  Output  Print table |
|  |

Class

|  |
| --- |
| //single class  #include<iostream>  using namespace std;  //declare and inisilization  // function declare and define insise a class  class top{      int a,b,c;      public:void sum(int x,int y)      {          a=x;          b=y;          c=a+b;      }  void show(){      cout<<"Result = "<<c;  }  };  int main(){      top t;      int s,b;      cout<<"Enter two Number \n";      cin>>s>>b;      t.sum(s,b);      t.show();  }  Output  Enter two Number  10  20  Result = 30  #include <iostream>  using namespace std;  // function declare insise a class but  define outside  a class  class top  {      int a, b, c;  public:      void sum(int x, int y);      void show();  };  //outside function  void top::sum(int x, int y)  {      a = x;      b = y;      c = a + b;  };  void top::show()  {      cout << "Result = " << c;  };  int main()  {      top t;      int s, b;      cout << "Enter two Number \n";      cin >> s >> b;      t.sum(s, b);      t.show();  }  output  Enter two Number  10  20  Result = 30 |
| #include <iostream>  using namespace std;  // function declare insise a class but  define outside  a class  class top  {      int a, b, c;  public: void sum(int a, int b)  {      a = a;      b = b;      c = a + b;  }  void show()  {      cout <<a<<" + "<<b<<" = " << c;  }  };  int main()  {      top t;      int s, b;      cout << "Enter two Number \n";      cin >> s >> b;      t.sum(s, b);      t.show();  }  Output  Enter two Number  10  20  4200992 + 6422352 = 30 |
| // this keywprd use  #include <iostream>  using namespace std;  class top  {      int a, b, c;  public: void sum(int a, int b)  {     this-> a = a;     this-> b = b;      c = a + b;  }  void show()  {      cout <<a<<" + "<<b<<" = " << c;  }  };  int main()  {      top t;      int s, b;      cout << "Enter two Number \n";      cin >> s >> b;      t.sum(s, b);      t.show();  }  Enter two Number  10  20  10 + 20 = 30 |
|  |

Function Overloading

|  |
| --- |
| #include <iostream>  using namespace std;  class top {  public:  // Functions  void sum() {  cout << "hello\n";  }  void show() {  cout << "Show function\n";  }  void display() {  cout << "display function\n";  }  // Function overloading  void sum(int a, int b) {  cout << "function 1 overload \n";  cout << "sum = " << a + b << endl;  }  void sum(float a, float b) {  cout << "function 2 overload \n";  cout << "sum = " << a + b << endl;  }  void sum(double a, double b) {  cout << "function 3 overload \n";  cout << "sum = " << a + b << endl;  }  };  int main() {  top k;  k.sum(); // Calls sum() with no arguments  k.sum(10, 20); // Calls sum(int, int)  k.sum(10.5f, 20.5f); // Calls sum(float, float)  k.sum(10.123, 20.456); // Calls sum(double, double)  k.show(); // Calls show()  k.display(); // Calls display()  return 0;  } |
| hello  function 1 overload  sum = 30  function 2 overload  sum = 31  function 3 overload  sum = 30.579  Show function  display function |
|  |
|  |

Function overriding

|  |
| --- |
|  |
|  |
|  |

Static keyword

|  |
| --- |
| #include<iostream>  using namespace std;  class cybrom{      public:static int a,b,c;      public:static void sum(int x,int y){          c=x+y;          cout<<c;      }  };  int cybrom::c=0; //static variable must be outside the class inisilize  int main(){      int x,y;      cout<<"Enter two Number\n";      cin>>x>>y;      cybrom ::sum(x,y);  } |
| Output  10  20  30 |

|  |
| --- |
| #include<iostream>  using namespace std;  class cybrom  {      public:void show()      {          cout<<"Cybrom class \n";      }  };  class bhopal{      public:void display(){      cybrom c;          c.show();      cout<<"Bhopal class\n";      }  };  int main(){     bhopal b;     b.display();  }  Output  Cybrom class  Bhopal class |
|  |

Friend function

|  |
| --- |
| // friend function is use to acces the private data memeber or memeber function of any class. for that we have to declare friends function  // in both classes. for friend functio using friends keyword  #include <iostream>  using namespace std;  class Tenth;  // Forward declaration of class Tenth  class Twelve {      int a;  public:      void twelveResult(int p) {          a = p;      }      friend void show(Twelve, Tenth);  // Friend function declaration  };  class Tenth {      int b;  public:      void tenResult(int p) {          b = p;      }      friend void show(Twelve, Tenth);  // Friend function declaration  };  void show(Twelve t12, Tenth t10) {      if (t12.a > t10.b) {          cout << "Result: " << t12.a;      } else {          cout << "Result: " << t10.b;      }  }  int main() {      Twelve t12;      Tenth t10;      t12.twelveResult(97);      t10.tenResult(70);      show(t12, t10);      return 0;  }  Result: 97 |

Constructor

// constructor is called special function which is use to allocate the memory space of any object in run time there are following type of constructor

1 default contstructor

2 paremetrize constructor

3 copy constructor two apart

1) shallow coppy

1) deep coppy

3 construtor overloading

distructor is use to deallocate the memory or releize the memory of the obect.

by defualt  class having  constrotor and distructor

|  |
| --- |
| // default constrotor in c++ is the function name is similar to class name wihtout usig any parameter is callled default constrotor  // construtor does not have any type  /\*  public:classname()  {  statement;  }  \*/  // default constructor  #include<iostream>  using namespace std;  class student{      public:student(){      cout<<"memory allocated\n";      }      void hello()      {          cout<<"hello";      }  };  int main(){      student s; //implicit calling      s.hello(); //explicit calling  }  Output  Memory allocated  hello |
|  |
| //object ko array ki tarah store kar stakte hai  //we can use array as a object  #include <iostream>  using namespace std;  class student  {  public:      student(int a, int b)      {          cout << "Memory allcated a and b\n";      }      student()      {          cout << "\nconstructor overloading";      }  };  int main()  {      student d(4, 5);      student g(8, 5);      student r;      student k;  } |
| Memory allcated a and b  Memory allcated a and b  constructor overloading  constructor overloading |

|  |
| --- |
| //shalow copy by default bana hota hai  // c++ contains the shalow copy inside a class  //  #include<iostream>  using namespace std;  class top{      int x,y;      public:top(int a,int b)      {          x=a;          y=b;      }  void show()  {    cout<<"x= "<<x<<endl;    cout<<"y= "<<y<<endl;  }  };  int main(){    top p1(40,50);    //top p2=p1; //implicit assignment copy constructor    top p2 (p1);//call shallow copy constructor    p2.show();  } |
|  |