**C++ static**

In C++, static is a keyword or modifier that belongs to the type not instance. So instance is not required to access the static members. In C++, static can be field, method, constructor, class, properties, operator and event.

**Advantage of C++ static keyword**

**Memory efficient:** Now we don't need to create instance for accessing the static members, so it saves memory. Moreover, it belongs to the type, so it will not get memory each time when instance is created.

**C++ Static Field**

A field which is declared as static is called static field. Unlike instance field which gets memory each time whenever you create object, there is only one copy of static field created in the memory. It is shared to all the objects.

It is used to refer the common property of all objects such as rateOfInterest in case of Account, companyName in case of Employee etc.

**C++ static field example**

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  class Employee {  public:  int id;  string name;  float salary;  static float cocksucker;  Employee(int i, string n, float s)  {  this->id = i;  this->name = n;  this->salary = s;  }  void display()  {  cout << "fucker's id : " << id << " fucker's name : " << name << " Cocksucker : " << cocksucker << endl;  }  };  //static is a common fuckin property  float Employee :: cocksucker = 6.5;  int main(void) {  Employee e1 =Employee(101, "white walkers", 890000);  Employee e2=Employee(102, "snow", 59000);  e1.display();  e2.display();  return 0;  }  **/\*fucker's id : 101 fucker's name : white walkers Cocksucker : 6.5**  **fucker's id : 102 fucker's name : snow Cocksucker : 6.5\*/** | #include <iostream>  using namespace std;  class Employee {  public:  int id;  string name;  float salary;  static int cocksucker;  Employee(int i, string n, float s)  {  this->id = i;  this->name = n;  this->salary = s;  cocksucker++;  }  void display()  {  cout << "fucker's id : " << id << " fucker's name : " << name << " Cocksucker : " << cocksucker << endl;  }  };  //static is a common fuckin property  // COUNTING BASTERDS  int Employee :: cocksucker = 0;  int main(void) {  Employee e1 =Employee(101, "white walkers", 890000);  Employee e2=Employee(102, "snow", 59000);  e1.display();  e2.display();  return 0;  }  **/\*fucker's id : 101 fucker's name : white walkers Cocksucker : 2**  **fucker's id : 102 fucker's name : snow Cocksucker : 2\*/** |