|  |  |
| --- | --- |
| Programs | Output |
| **//Single Inheritance**  #include<iostream>  using namespace std;  class A  {protected:  int x,y;  public: void input()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class B:public A  {public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output1()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  int main()  {B obj;  cout<<"\nFunction call to the derived class.";  obj.input1();  obj.output1();  cout<<"\nFunction call to the base class.";  obj.input();  obj.output();  return 0;} |  |
| **//Hybrid Inheritance**  #include<iostream>  using namespace std;  class A  {protected: int x,y;  public: void input()  { cout<<"\nEnter two numbers: ";  cin>>x>>y; }  void output()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class B:public virtual A  {public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output1()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class C:public virtual A  {public: void input2()  { cout<<"\nEnter two numbers: ";  cin>>x>>y; }  void output2()  { cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class D:public B,public C  {public: void input3()  { cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output3()  { cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  int main()  {D obj;  cout<<"\nIn derived class D :\n";  obj.input3();  obj.output3();  cout<<"\nIn derived class C :\n";  obj.input2();  obj.output2();  cout<<"\nIn derived class B :\n";  obj.input1();  obj.output1();  cout<<"\nIn base class A:\n";  obj.input();  obj.output();  return 0;  } |  |
| **//Hierarchical Inheritance**  #include<iostream>  using namespace std;  class A  {  protected:  int x,y;  public: void input(){  cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class B:public A  {public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output1()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class D:public B  {public: void input3()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output3()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class E:public B  {public: void input4()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output4()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class C:public A  {public: void input2()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output2()  {cout<<"The two numbers are "<<x<<" and "<<y<<".\n";}  };  class F:public C  {public: void input5()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output5()  {cout<<"The two numbers are "<<x<<" and "<<y<<".\n";}  };  class G:public C  {int x,y;  public: void input6()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output6()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  int main()  {D one; E two; F three; G four;  cout<<"\nDerived class D.";  one.input3();one.output3();one.input1();  one.output1();one.input();one.output();  cout<<"\nDerived class E.";  two.input4();two.output4();two.input1();  two.output1();two.input();two.output();  cout<<"\nDerived class F.";  three.input5();three.output5();three.input2();  three.output2();three.input();three.output();  cout<<"\nDerived class G.";  four.input6();four.output6();four.input2();  four.output2();four.input();four.output();  return 0;} |  |
| **//Hybrid Inheritance**  #include<iostream>  using namespace std;  class A  {protected:int x,y;  public: void input()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class B  {protected:int x,y;  public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output1()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class C:public A,public B  {protected:int x,y;  public: void input2()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output2()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class D:public C  {  public: void input3()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output3()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  class E:public C  {public: void input4()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output4()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";}  };  int main()  {  D one; E two;  cout<<"\nDerived class D.:\n";  one.input3();one.output3();  one.input2();one.output2();  one.input1();one.output1();  one.input();one.output();  cout<<"\nDerived class E.:\n";  two.input4();two.output4();  two.input2();two.output2();  two.input1();two.output1();  two.input();two.output();  return 0;  } |  |
| **//Multilevel Inheritance**  #include<iostream>  using namespace std;  class A  {int x,y;  public: void input()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output()  {cout<<"\nThe two numbers are"<<x<<" and "<<y<<".\n";  }  };  class B:public A  {int x,y;  public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;  }  void output1()  {cout<<"\nThe two numbers are"<<x<<" and "<<y<<".\n";  }  };  class C:public B  {int x,y;  public: void input2()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output2()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";  }  };  int main()  {C one;  cout<<"\nFunction call of derived class.\n";  one.input2();  one.output2();  cout<<"\nFunction call of intermediate class.\n";  one.input1();  one.output1();  cout<<"\nFunction call of base class.\n";  one.input();  one.output();  return 0;  } |  |
| **//Multiple Inheritance**  #include<iostream>  using namespace std;  class A  {int x,y;  public: void input()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;}  void output()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";  }  };  class B  {int x,y;  public: void input1()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;  }  void output1()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";  }  };  class C:public A,public B  {int x,y;  public: void input2()  {cout<<"\nEnter two numbers: ";  cin>>x>>y;  }  void output2()  {cout<<"\nThe two numbers are "<<x<<" and "<<y<<".\n";  }  };  int main()  {C one;  cout<<"\nFunction calls of derived class.\n";  one.input2();  one.output2();  cout<<"\nFunction calls of base class two.\n";  one.input1();  one.output1();  cout<<"\nFunction calls of base class one.\n";  one.input();  one.output();  return 0;  } |  |