```
# include < iostream.h >
# include < conio.h >
 class prime
{
 int n, i, flag ;
 public :
  void check ()
 {
  flag = 0 ;
  cout << " \n entre any number " ;
  cin >> n ;
 for ( i=2 ; i<n ; i++ )
 {
  if ( n % i == 0 )
 {
  flag = 1 ;
  break ;
 }
 }
 if ( flag == 1 )
 cout << " \n number is not prime " ;
 else
 cout << " \n number is prime " ;
 }
};
 void main ()
 {
  clrscr ();
  prime obj ;
  obj . check () ;
  getch ();
 }
```

```
# include < iostream.h >
# include < conio.h >
 void swap ( int x, int y )
{
 int t;
cout << " before swaping = \t " << x << " \t " << y ;
 t = x ;
 x = y ;
 y = t ;
cout << " \n after swaping = \t" << x << " \t" << y ;
}
 void swap ( float x , float y )
{
 float t ;
cout << " before swaping \t " << x << " \t " << y ;
 t = x ;
 x = y ;
 y = t ;
cout << " \n after swaping \t" << x << " \t " << y ;
}
 void main ()
{
 float ch1 , ch 2 ;
 int a , b ;
 clrscr ();
cout << " \n entre the two float no " ;
cin >> ch1 >> ch2 ;
swap ( ch1 , ch 2 );
cout << " \n Entre the two no. \n " ;
 cin >> a >> b ;
swap ( a , b );
  getch ();
}
```

```
# include < iostream . h >
# include < conio . h >
 class demo
{
 public :
    int a , b , c ;
     demo ()
  {
   a = 10 ;
   b = 20 ;
  }
   void print ()
 {
  cout << " Hello , Good Morning " ;
  cout << " \n a= " << a << " \n b = " << b ;
  }
 } ;
   void main ()
 {
   clrscr () ;
   demo obj ;
   obj . print () ;
    getch () ;
 }
```

```
# include < iostream.h >
# include < conio.h >
 class cons
{
  public :
     cons ()
{
 cout << " This is the constructor function " ;
 cout << " This function is used to initialise
                the variable " ;
}
~ cons ()
 {
 cout << " \n This is the Destructor function \n " ;
 cout << " This is used to destroy the memory
 }
};
 void main ()
 {
  clrscr ();
  cons c ;
  c. ~cons ();
  getch ();
 }
```

```
# include < iostream.h >
# include < conio.h >
  class A
{
  int a,b;
  void getdata ()
{
  cout << "\n entre value of a & b";
   cin >> a >> b;
  }
};
  class B : public A
  {
  int c,d;
  public : void cal ()
{
  c = a+b ;
  cout << "\n addition is " << c;
  d = a-b;
  cout << "\n substraction is " << d;
 }
};
 void main ()
{
  clrscr ();
   B obj;
  obj.getdata ();
  obj. cal ();
   getch ();
 }
```

```
# include <iostream.h>
# include <conio.h>
  class A
{
  public : void show1 ()
   {
    cout << " This is the Base 1 class \n";
   }
};
   class B
   {
   public : void show2 ()
    {
     cout << " This is the Base 2 class \n";
    }
  };
  class C : public A, public B;
  {
   public : void show 3 ()
  {
   cout << " This is the Derived from class A
                & class B \n";
   }
};
  void main ()
{
  clrscr ();
  C obj;
  obj. show1 ();
  obj. show2 ();
  obj. show3 ();
  getch ();
}
```

```
# include <iostream.h>
# include <conio.h>
 class complex
{
  int a,b;
 public : void getvalue ()
  {
   cin >> a >> b;
  }
  void display ()
  {
   cout << a << " + " << b << "i" << "\n";
  }
  friend complex operator + ( complex obp ,
                              complex ob );
};
 complex operator + ( complex obp, complex ob
 {
  complex t;
  t.a = obp.a + ob.a;
  t.b = obp.b + ob.b;
  return (t);
 }
void main ()
{
  clrscr ();
  complex obj1, obj2, res;
  cout << "\n enter 1st complex no.";
  obj1.getvalue ();
  cout << "\n enter 2nd complex no.";
  obj2.getvalue ();
  res = obj1 + obj2;
  cout << "\n Result ";
  res.display ();
  getch ();
```

```
# include <iostream.h>
# include <conio.h>
 class complex
{
  int a,b;
 public : void getvalue ()
  {
   cin >> a >> b;
  }
 complex operator + ( complex ob )
 {
  complex t;
  t.a = a + ob.a;
  t.b = b + ob.b;
  return (t);
 } void
 void display ()
 {
  cout << "complex number is = \t";
  cout << a << " + " << b << " i " << " \n";
 }
};
 void main ()
{
 clrscr ();
 complex obj1, obj2, res;
 cout << "\n Entre 1st complex no \n";
 obj1.getvalue ();
 obj1.display ();
  cout << " \n entre 2nd complex no.";
 obj2.getvalue ();
 obj2.display ();
 res = obj1 + obj2;
 cout << " \n Result = ";
 res.display ();
 getch ();
}
```

```
# include < iostream.h >
# include < conio.h >
 class two;
 class one
{
  private :
    int data 1 ;
  public :
    void setdata ( int init )
   {
     data 1 = init ;
   }
  friend int add -- both ( one a , two b);
};
  class two
  {
    private :
       int data 2 ;
    public :
       void setdata ( int init )
   {
     data 2 = init ;
   }
 friend int add _ both ( one a , two b );
};
 int add both ( one a , two b)
 {
  return a . data 1 + b . data 2
 }
```

```
void main ()
{
 clrscr ();
 one a;
 two b;
 a. setdata (5);
 b. setdata (10);
 cout << " sum of one & two :- \t " <<
                    add - both (a,b);
 getch ();
}
```

```
# include < iostream . h >
# include < conio . h >
 class boy
{
 private :
   int income 1 ;
   int income 2 ;
public :
   void setdata ( int in1 , int in2 )
{
   income 1 = in1 ;
   income 2 = in2 ;
}
 friend class girl ;
};
   class girl
{
int income ;
 public :
   int girl func ( boy b1 )
{
 return b1 . income 1 + b1 . income 2 ;
}
   void setdata ( int in )
{
   income = in ;
}
   void show ()
{
   boy b1 ;
   b1 . setdata ( 100 , 200 );
```

```
cout << " boys Income 1 in show () : \t "
        << b1.income1 << endl;
cout << girls Income 2 in show () : \t "
        << income << endl;
  }
};
 void main ()
{
 clrscr ();
 boy b1;
 girl g1;
 b1.setdata (500, 1000);
 g1.setdata (300);
 cout << " boy b1 total income : \t " <<
        g1.girl func (b1) << endl;
  g1.show ();
  getch ();
}
```

```
# include < iostream.h >
# include < conio.h >
 class stud
{
 public : virtual void display ()
  {
     cout << "\n student data will be displayed "
  }
};
 class college : public stud
  {
     public : void display ()
      {
       cout << "\n college data ";
      }
  };
 void main ()
  {
   clrscr ();
   stud s;
   s.display ();
   stud *p;
   college obj;
   p = & obj;
   p -> display ();
   getch ();
  }
```

```
#include <iostream.h>
#include <conio.h>
template <class T>
 T min (T a, T b)
 {
  if (a<b)
   return a;
  else
    return b;
  }
void main ()
  {
   clrscr ();
   int i,j;
   cout << "\n Enter integer values ";
   cin >> i >> j;
   cout << endl << " min of two integers = " <<
                                  min (i,j);
   float a,b;
   cout << "\n Enter float values ";
   cin >> a >> b;
   cout << endl << " min of two float numbers =
          "<<min (a,b);
    getch ();
}
```