فصل هشتم: آشنایی با کلاس‌ها

```
// S08-01.cpp
// استفاده از کلاس برای مقدار دهی و چاپ زمان

#include "stdafx.h"
#include "iostream"

using namespace std;

class time
{
 public:
  void set_time(int,int,int);
  void print_time();
 private:
  int hour;
  int minute;
  int second;
} t1;

void time::set_time(int h,int m,int s)
{
 hour=h;
 minute=m;
 second=s;
}

void time::print_time()
{
 cout<<"Time is : ";
 cout<<hour<<":"<<minute<<":"<<second;
}

int main(void)
{
 int h,m,s;
 cout<<"Enter hour : ";
 cin>>h;
 cout<<"Enter minute : ";
 cin>>m;
 cout<<"Enter second : ";
 cin>>s;
 t1.set_time(h,m,s);
 t1.print_time();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter hour : 12
Enter minute : 30
Enter second : 25
Time is : 12:30:25
Press any key to continue . . .
```

```
// S08-02.cpp
// محاسبه محیط و مساحت دایره با استفاده از کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class circle
{
 float radius;
public:
 void get_radius(void);
 void print_area(void);
 void print_perime(void);
};

void circle::get_radius(void)
{
 cout<<"Enter radius : ";
 cin>>radius;
}

void circle::print_area(void)
{
 double a;
 a=radius*radius*3.14;
 cout<<"\nArea is : "<<a;
}

void circle::print_perime(void)
{
 double p;
 p=2*radius*3.14 ;
 cout<<"\nPerime is : "<<p;
}

int main(void)
{
 circle c ;
 c.get_radius();
 c.print_area();
 c.print_perime();
 cout<<endl;
 system("pause");
 return 0;
}
```

Enter radius : 3

Area is : 28.26
Perime is : 18.84
Press any key to continue . . .

```
// S08-03.cpp
// محاسبه محیط و مساحت دایره با استفاده از سازنده کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class circle
{
 float radius;
public :
 circle(void);
 void print_area(void);
 void print_perime();
};

circle::circle(void)
{
 radius=3;
}

void circle::print_area(void)
{
 double a;
 a=radius*radius*3.14;
 cout<<"\nArea is : "<<a;
}

void circle::print_perime(void)
{
 double p;
 p=2*radius*3.14;
 cout<<"\nPerime is : "<<p;
}

int main(int argc, char* argv[])
{
      circle c ;
      c.print_area();
      c.print_perime();
    cout<<endl;
    system("pause");
    return 0;
}
```

```
Area is : 28.26
Perime is : 18.84
Press any key to continue . . .
```

```
// S08-04.cpp
// نمایش سازنده و مخرب کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
public:
 Test(void);
 ~Test(void);
};

Test::Test(void)
{
 cout<<"Runing Constructor. \n";
}

Test::~Test(void)
{
 cout<<"Runing Destructor. \n";
}

int main(void)
{
 cout<<"Before Define Object. \n";
 Test T;
 cout<<"After Define Object. \n";
 cout<<endl;
 system("pause");
 return 0;
}
```

Before Define Object.
Runing Constructor.
After Define Object.

Press any key to continue . . .
Runing Destructor.

```
// P08-01.cpp
// نحوه دسترسی به اعضای خصوصی، محافظت شده و عمومی کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;


class MyClass
{
 int a;
protected:
 int b;
public:
 int c;
 MyClass(int n,int m)
 {
  a=n;
  b=m;
  }
 int geta(void)
 {
  return a;
 }
 int getb(void)
 {
  return b;
 }
};

int main(void)
{
 MyClass object(10,20);
 object.c=30;
 cout<<object.geta()<<" ";
 cout<<object.getb()<<" "<<object.c<<"\n";
 cout<<endl;
 system("pause");
 return 0;
}
```

```
10 20 30
Press any key to continue . . .
```

```
// P08-02.cpp
// چاپ ستاره به تعداد مقدار یک عضو داده اش

#include "stdafx.h"
#include "iostream"

using namespace std;


class SimpleClass
{
 int len;
public:
 SimpleClass(int l);
};

SimpleClass::SimpleClass(int l)
{
 len=l;
 int i;
 for(i=0;i<len;i++)
  cout<<"*";
 }

int main(void)
{
 SimpleClass A(10);
 cout<<endl;
 system("pause");
 return 0;
}
```

```
**********
Press any key to continue . . .
```

```
// P08-03.cpp
// محاسبه توان با استفاده از کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;


class Power
{
 double b;
 int e;
 double val;
public:
 Power(double base,int exp);
 double getPower()
 {
  return val;
 }
};

Power::Power(double base,int exp)
{
 b=base;
 e=exp;
 val=1;
 if(exp==0)
  return;
 for(;exp>0;exp--)
  val=val*b;
}

int main(void)
{
 Power x(4.0 , 2),y(2.5 , 1),z(5.7 , 0);
 cout<<x.getPower()<<"  ";
 cout<<y.getPower()<<"  ";
 cout<<z.getPower()<<endl;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
16  2.5  1

Press any key to continue . . .
```

```
// P08-04.cpp
// کلاس با ۲ سازنده

#include "stdafx.h"
#include "iostream"

using namespace std;


class myclass
{
 int a;
public:
 myclass(int x)
 {
  a=x;
 }
 myclass(char *str)
 {
  a=atoi(str);
 }
 int geta()
 {
  return a;
 }
};

int main(void)
{
 myclass object1=4;
 myclass object2="123";
 cout<<"object1: "<<object1.geta()<<endl;
 cout<<"object2: "<<object2.geta()<<endl;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
object1: 4
object2: 123

Press any key to continue . . .
```

```
// P08-05.cpp
// محاسبه حقوق کارمند با استفاده از کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;


class employee
{
 char name[50];
 int salary,hour,hp;
public:
 void input(void);
 void calc(void);
 void print(void);
};

void employee::input(void)
 {
 cout<<"Enter name,hours,hourpay : ";
 cin>>name>>hour>>hp;
 }

void employee::calc(void)
 {
 salary=hour*hp;
 }

void employee::print(void)
{
 cout<<"Salary is : "<<salary;
}

int main(void)
{
 employee e;
 e.input();
 e.calc();
 e.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter name,hours,hourpay : Hooman 175 4000
Salary is : 700000
Press any key to continue . . .
```

```
// P08-06.cpp
// کلاسی برای آرایه ای از اعداد با توانائی ورود، چاپ، جستجو و مرتب سازی آرایه

#include "stdafx.h"
#include "iostream"
using namespace std;

class ArrayInt
{
 int A[10];
public:
 void Input(void);
 void Output(void);
 void Sort(void);
 int Search(int key);
 int Max(void);
};

void ArrayInt::Input(void)
{
 cout<<"Enter 10 numbers : ";
 for(int i=0;i<10;i++)
  cin>>A[i];
}

void ArrayInt::Output(void)
{
 cout<<"Numbers in Array are : ";
 for(int i=0;i<10;i++)
  cout<<A[i]<<" ";
 cout<<"\n";
}

void ArrayInt::Sort(void)
{
 int i,j,t;
 for(i=0;i<10;i++)
  for(j=i+1;j<10;j++)
   if(A[i]>A[j])
   {
    t=A[i];
    A[i]=A[j];
    A[j]=t;
   }
}

int ArrayInt::Search(int key)
{
int i;
for(i=0;i<10;i++)
 if(A[i]==key)
  return i;
return 0;
}

int ArrayInt::Max(void)
{
int i,m;
m=A[0];
for(i=1;i<10;i++)
 if(A[i]>m)
  m=A[i];
return(m);
}
```

```
int main(void)
{
 int key,f;
 ArrayInt A;
 A.Input();
 A.Sort();
 A.Output();
 cout<<"Maximum number is : "<<A.Max()<<"\n";
 cout<<"Enter One number for search : ";
 cin>>key;
 f=A.Search(key);
 if(f==0)
  cout<<"Not Find.";
 else
  cout<<"Find.";
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter 10 numbers : 4 5 6 8 5 2 1 5 9 7
Numbers in Array are : 1 2 4 5 5 5 6 7 8 9
Maximum number is : 9
Enter One number for search : 2
Find.
Press any key to continue . . .
```

```
// S09-01.cpp
// اشیا و توابع ثابت در کلاس

##include "stdafx.h"
#include "iostream"
#include <iomanip>

using namespace std;

class Time
{
public:
 Time(int=0,int=0,int=0);
 void setTime(int,int,int);
 void setHour(int);
 void setMinute(int);
 void setSecond(int);
 int getHour()const;
 int getMinute() const;
 int getSecond() const;
 void printUniversal() const;
 void printStandard();
private:
 int hour;
 int minute;
 int second;
};

Time::Time(int hour,int minute,int second)
{
 setTime(hour,minute,second);
}

void Time::setTime(int hour,int minute,int second)
{
 setHour(hour);
 setMinute(minute);
 setSecond(second);
}

void Time::setHour(int h)
{
 hour=(h>=0&&h<24)?h:0;
}

void Time::setMinute(int m)
{
 minute=(m>=0&&m<60)?m:0;
}

void Time::setSecond(int s)
{
 second=(s>=0&&s<60)?s:0;
}

int Time::getHour() const
{
 return hour;
}

int Time::getMinute() const
{
 return minute;
}
```

```
int Time::getSecond() const
{
 return second;
}

void Time::printUniversal() const
{
 cout<<"\n"<<setfill('0')<<setw(2)<<hour<<":"<<setw(2);
 cout<<minute<<":"<<setw(2)<<second;
 }

void Time::printStandard()
{
 cout<<"\n"<<((hour==0||hour==12)?12:hour%12)<< ":";
 cout<<setfill('0')<<setw(2)<<minute<< ":";
 cout<<setw(2)<<second<<(hour<12?" AM":" PM");
 }

int main(void)
{
 int h,m;
 Time Wakeup(6,45,25);
 const Time noon(12,0,0);
 Wakeup.setHour(18);
 h=Wakeup.getHour();
 cout<<"h= "<<h;
 Wakeup.printStandard();
 Wakeup.printUniversal();
 cout<<"\n"<<"\n";
 m=noon.getMinute();
 cout<<"m= "<<m;
 //noon.setHour(17);       Error: "noon" is Const
 //noon.printStandard();   Error: "noon" is Const
 noon.printUniversal();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
h= 18
6:45:25 PM
18:45:25

m= 0
12:00:00
Press any key to continue . . .
```

```
// S09-02.cpp
// تغییر مقدار داده کلاس با یک تابع ثابت

#include "stdafx.h"
#include "iostream"

using namespace std;

class Mclass
{
mutable int x;
public:
 Mclass(int i)
 {
  x=i;
 }
 void incX(void) const
 {
  x++;
 }
 void print(void)
 {
  cout<<"\nx = "<<x;
 }
}obj(3);

int main(void)
{
 obj.print();
 obj.incX();
 cout<<"\nafter increment : ";
 obj.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x = 3
after increment :
x = 4
Press any key to continue . . .
```

```
// S09-03.cpp
// عضو داده ای ثابت در کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class Increment
{
public:
 Increment(int c=0,int i=1);
 void addIncrement()
 {
   count+=increment;
 }
 void print() const;
private:
 int count;
 const int increment;
};

Increment::Increment(int c,int i):count(c),increment(i)
{
}

void Increment::print() const
{
  cout<<"count="<<count<<" , increment=";
  cout<<increment<<endl;
}

int main(void)
{
 Increment value(10,5);
 cout<<"Before incrementing:";
 value.print();
 for(int j=1;j<=3;j++)
 {
  value.addIncrement();
  cout<<"After increment:"<<j<< ": ";
  value.print();
 }
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Before incrementing:count=10 , increment=5
After increment:1: count=15 , increment=5
After increment:2: count=20 , increment=5
After increment:3: count=25 , increment=5

Press any key to continue . . .
```

```
// S09-04.cpp
// تـابـع دوست

#include "stdafx.h"
#include "iostream"

using namespace std;

class Count
{
friend void setX(Count & , int);
public:
 Count()
 {
  x=0;
 }
 void print() const
 {
  cout<<x<<endl;
 }
private:
 int x;
};

void setX(Count &c,int val)
{
 c.x=val;
}

int main(void)
{
 Count c;
 cout<<"c.x after instantiation:";
 c.print();
 setX(c,2);
 cout<<"c.x after call setX friend function:";
 c.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
c.x after instantiation:0
c.x after call setX friend function:2
Press any key to continue . . .
```

```
// S09-05.cpp
// تابع دوست

#include "stdafx.h"
#include "iostream"
using namespace std;

class ScoreClass
{
 double MintermScore,FinaltermScore;
 friend void check(ScoreClass s);
public:
 void input(double,double);
};

void ScoreClass::input(double m,double f)
{
 MintermScore=m;
 FinaltermScore=f;
}

void check(ScoreClass s)
{
 int m=0;
 double sc;
 if((s.MintermScore>=0)&&(s.MintermScore<=5))
   cout<<"MintermScore = "<<s.MintermScore;
 else
  {
   cout<<"MintermScore is mistake. ";
   m = 1;
  }
 cout<<endl;
 if((s.FinaltermScore>=0)&&(s.FinaltermScore<=15))
   cout<<"FinaltermScore = "<<s.FinaltermScore;
 else
  {
   cout<<"MintermScore is mistake. ";
   m = 1 ;
  }
 cout<<endl;
 if(m==1)
  cout<<"FinalScore is mistake. ";
 else
  {
   sc=s.FinaltermScore+s.MintermScore;
   cout<<"FinalScore = "<<sc;
  }
}

int main(void)
{
 ScoreClass score;
 double m,f;
 cout<<"Enter Minterm Score : " ;
 cin>>m;
 cout<<"Enter Finalterm Score : " ;
 cin>>f;
 score.input(m,f);
 check(score);
 cout<<endl;
 system("pause");
 return 0;
```

```
}
```

```
Enter Minterm Score : 4
Enter Finalterm Score : 13
MintermScore = 4
FinaltermScore = 13
FinalScore = 17
Press any key to continue . . .
```

```
// S09-06.cpp
// کلاس دوست

#include "stdafx.h"
#include "iostream"

using namespace std;

class employee
{
 friend class TaxSalary;
 char name[50];
 double s,hour,hp,salary;
public:
 void input(void);
 void calc(void);
 void print(void);
};

void employee::input(void)
 {
 cout<<"Enter name : ";
 cin>>name;
 cout<<"Enter hours : ";
 cin>>hour;
 cout<<"Enter hourpay : ";
 cin>>hp;
 }

void employee::calc(void)
 {
 s=hour*hp;
 }

void employee::print(void)
{
 cout.setf(ios::fixed);
 cout<<"Salary is : "<<salary;
}

class TaxSalary
{
 double tax;
public:
 void input(void)
 {
  cout<<"Enter percent tax : ";
  cin>>tax;
 }

 void CalcFinalSalary(employee &e)
 {
       e.salary=e.s-(tax*e.s/100);
 }
};


int main(void)
{
 employee e;
 e.input();
 e.calc();
 TaxSalary t;
 t.input();
```

```
 t.CalcFinalSalary(e);
 e.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter name : Hooman
Enter hours : 175
Enter hourpay : 5000
Enter percent tax : 10
Salary is : 787500.000000
Press any key to continue . . .
```

```
// S09-07.cpp
// مقایسه عضو داده ای استاتیک با غیر استاتیک

#include "stdafx.h"
#include "iostream"

using namespace std;

class Sclass
{
 static int x;
 int y;
public:
 void set(int i , int j);
 void print(void);
};
int Sclass::x;


void Sclass::set(int i,int j)
{
 x=i;
 y=j;
}

void Sclass::print(void)
{
 cout<<"\nStatic x = "<<x;
 cout<<"\nNon Static y = "<<y;
}

int main(void)
{
 Sclass obj1,obj2;
 obj1.set(4,4);
 obj1.print();
 obj2.set(5,5);
 obj2.print();
 obj1.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Static x = 4
Non Static y = 4
Static x = 5
Non Static y = 5
Static x = 5
Non Static y = 4
Press any key to continue . . .
```

```
// S09-08.cpp
// شمارش تعداد اشیا یک کلاس با استفاده از یک عضو داده ای استاتیک

#include "stdafx.h"
#include "iostream"

using namespace std;

class counter
{
public:
 static int count;
 counter(void)
 {
  count++;
 }
};
int counter::count;

int main(void)
{
 counter c1,c2,c3;
 cout<<"Numbers of object is : ";
 cout<<counter::count;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Numbers of object is : 3
Press any key to continue . . .
```

```
// S09-09.cpp
// استفاده سراسری از یک متغیر استاتیک کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class Sclass
{
public:
 static int x;
 void print(void)
 {
  cout<<"x = "<<x;
 }
};

int Sclass::x;
int main(void)
{
 Sclass::x = 10;
 Sclass object;
 object.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x=10
Press any key to continue . . .
```

```
// S09-10.cpp
// مقدار دهی یک داده استاتیک کلاس با یک تابع استاتیک

#include "stdafx.h"
#include "iostream"

using namespace std;

class sclass
{
static int i;
public:
 static void init(int x)
  {
   i=x;
  }
 void show(void)
 {
  cout<<"i="<<i;
 }
};
int sclass::i;

int main(void)
{
 sclass::init(100);
 sclass s;
 s.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
i=100
Press any key to continue . . .
```

```
// S09-11.cpp
// چاپ محتویات یک کلاس با یک تابع

#include "stdafx.h"
#include "iostream"

using namespace std;

class stud
{
public:
 int id;
 char name[50];
 char family[50];
 void input(void);
};

void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}

void PrintObject(stud s)
{
 cout<<"\nID = "<<s.id;
 cout<<"\nName = "<<s.name;
 cout<<"\nFamily = "<<s.family;
}

int main(void)
{
 stud s1;
 s1.input();
 PrintObject(s1);
 cout<<endl;
 system("pause");
 return 0;
}
```

Enter Id,Name,Family : 110 Hooman Sayyari

ID = 110
Name = Hooman
Family = Sayyari
Press any key to continue . . .

```
// S09-12.cpp
// تابعی که دو شی از یک کلاس را میگیرد و بزرگتر را چاپ میکند

# #include "stdafx.h"
#include "iostream"

using namespace std;

class stud
{
public:
 int id;
 char name[50];
 char family[50];
 void input(void);
 void print(void);
};

void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}

void stud::print(void)
{
 cout<<"\nID = "<<id;
 cout<<"\nName = "<<name;
 cout<<"\nFamily = "<<family;
}

stud  MaxID(stud o1,stud o2)
{
 if (o1.id>o2.id)
  return o1;
 else
  return o2;
}

int main(void)
{
 stud s1,s2,s;
 s1.input();
 s2.input();
 s=MaxID(s1,s2);
 cout<<"\nObject With Maximum id is : ";
 s.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter Id,Name,Family : 110 Hooman Sayyari
Enter Id,Name,Family : 220 Ali Khorrami

Object With Maximum id is :
ID = 220
Name = Ali
Family = Khorrami
Press any key to continue . . .
```

```
// S09-13.cpp
// تابعی که یک شی را گرفته و مقادیر آن را تغییر میدهد

#include "stdafx.h"
#include "iostream"

using namespace std;

class stud
{
public:
 int id;
 char name[50];
 char family[50];
 void input(void);
 void print(void);
};

void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}

void stud::print(void)
{
cout<<"\nID = "<<id;
cout<<"\nName = "<<name;
cout<<"\nFamily = "<<family;
}

void change(stud &s,char str1[50],char str2[50])
{
 strcpy(s.name,str1);
 strcpy(s.family,str2);
}

int main(void)
{
 char str1[50],str2[50];
 stud s1;
 s1.input();
 cout<<"Enter Name for change : ";
 cin>>str1;
 cout<<"Enter Family for change : ";
 cin>>str2;
 change(s1,str1,str2);
 s1.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

Enter Id,Name,Family : 110 Hooman Sayyari
Enter Name for change : Ali
Enter Family for change : Khorrami

ID = 110
Name = Ali
Family = Khorrami
Press any key to continue . . .

```
// S09-14.cpp
// آرایه ای از اشیا

#include "stdafx.h"
#include "iostream"
using namespace std;

class stud
{
 int id;
 char name[50];
 char family[50];
public:
 void input(void);
 void print(void);
};

stud s[5];
void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}
void stud::print(void)
{
 cout<<"\nID = "<<id;
 cout<<"\nName = "<<name;
 cout<<"\nFamily = "<<family;
}
int main(void)
{
 int i;
 for(i=0;i<5;i++)
  s[i].input();
 for(i=0;i<5;i++)
  s[i].print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter Id,Name,Family : 101 Ali Bahmani
Enter Id,Name,Family : 102 Reza Soleimani
Enter Id,Name,Family : 103 Dariush Dehroyeh
Enter Id,Name,Family : 104 Hooman Sayyari
Enter Id,Name,Family : 105 Pedram Ahmadi

ID = 101
Name = Ali
Family = Bahmani
ID = 102
Name = Reza
Family = Soleimani
ID = 103
Name = Dariush
Family = Dehroyeh
ID = 104
Name = Hooman
Family = Sayyari
ID = 105
Name = Pedram
Family = Ahmadi
Press any key to continue . . .
```

```
// S09-15.cpp
// آرایه ای از اشیا با سازنده یک پارامتری

#include "stdafx.h"
#include "iostream"

using namespace std;

class sample
{
 int x;
 public:
  sample(int a)
  {
   x=a;
  }
 int get(void)
  {
  return x;
  }
};

int main(void)
{
 int i;
 sample s[5]={3,4,5,6,7};
 for(i=0;i<5;i++)
  cout<<" s["<<i<<"]="<<s[i].get();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
s[0]=3 s[1]=4 s[2]=5 s[3]=6 s[4]=7
Press any key to continue . . .
```

```
// S09-16.cpp
// آرایه ای از اشیا با سازنده دو پارامتری

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
public:
int i , j;
Test(int a , int b)
 {
  i=a;
  j=b;
 }
void print(void)
{
 cout<<"i = "<<i<<" , j = "<<j;
}
};

int main(void)
{
 Test A[3]={Test(1,1),Test(2,2),Test(3,3)};
 for(int i=0;i<3;i++)
 {
  cout<<"\nobject "<<(i+1)<<" : ";
  A[i].print();
 }
 cout<<endl;
 system("pause");
 return 0;
}
```

```
object 1 : i = 1 , j = 1
object 2 : i = 2 , j = 2
object 3 : i = 3 , j = 3
Press any key to continue . . .
```

```
// S09-17.cpp
// آرایه ای از اشیا برای کلاسی با ۲ سازنده (با پارامتر و بدون پارامتر)

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
public:
int i , j;
Test(int a , int b)
 {
  i=a;
  j=b;
 }
Test(void)
{
 i=0;
 j=0;
}
void print(void)
{
 cout<<"i = "<<i<<" , j = "<<j;
}
};

int main(void)
{
 Test A[3]={Test(1,1),Test(2,2),Test(3,3)};
 cout<<"Array A :";
 for(int i=0;i<3;i++)
 {
  cout<<"\nobject "<<(i+1)<<" : ";
  A[i].print();
 }
 Test B[3];
 cout<<"\nArray B :";
 for(int j=0;j<3;j++)
 {
  cout<<"\nobject "<<(j+1)<<" : ";
  B[j].print();
 }
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Array A :
object 1 : i = 1 , j = 1
object 2 : i = 2 , j = 2
object 3 : i = 3 , j = 3
Array B :
object 1 : i = 0 , j = 0
object 2 : i = 0 , j = 0
object 3 : i = 0 , j = 0
Press any key to continue . . .
```

```
// S09-18.cpp
// اشاره گر به شی

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
 public:
 int i , j;
 void input(int a , int b)
 {
  i=a;
  j=b;
 }
};

int main(void)
{
 Test  obj,*p;
 p=&obj;
 obj.input(3,4);
 cout<<"obj.i = p->i = "<<p->i<<"  obj.i = (*p).i = "<<(*p).i;
 cout<<"\nobj.j = p->j = "<<p->j<<"  obj.j = (*p).j = "<<(*p).j;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
obj.i = p->i = 3  obj.i = (*p).i = 3
obj.j = p->j = 4  obj.j = (*p).j = 4
Press any key to continue . . .
```

```
// S09-19.cpp
// اشاره گر به آرایه ای از اشیا

#include "stdafx.h"
#include "iostream"

using namespace std;

class sample
{
 int x;
 public:
  sample(int a)
  {
   x=a;
  }
 int get(void)
  {
  return x;
  }
};

int main(void)
{
 sample s[5]={3,4,5,6,7};
 sample *p;
 p=s;
 for(int i=0;i<5;i++)
 {
  cout<<" s["<<i<<"]="<<p->get();
  p++;
 }
 cout<<endl;
 system("pause");
 return 0;
}
```

```
s[0]=3 s[1]=4 s[2]=5 s[3]=6 s[4]=7
Press any key to continue . . .
```

```
// S09-20.cpp
// چاپ اعضای یک شی با استفاده از اشاره گر this

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
public:
 int i , j;
 Test(int a , int b)
 {
  i=a;
  j=b;
 }
 void print(void)
 {
  cout<<"i = "<<this->i<<" , j = "<<this->j<<"\n";
  cout<<"i = "<<i<<" , j = "<<j;
 }
};

int main(void)
{
 Test  A(1,1);
 A.print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
i = 1 , j = 1
i = 1 , j = 1
Press any key to continue . . .
```

```
// S09-21.cpp
// اشاره گر به عضو کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class Test
{
public:
 Test(int i)
 {
  T=i;
 }
 void print(void)
 {
  cout<<"T = "<<T;
 }
 int T;
};

int main(void)
{
 int Test::*pd;
 void (Test::*pf)()=&Test::print;
 Test object1(4),object2(5);
 pd=&Test::T;
 cout<<"Values object : ";
 cout<<object1.*pd<<"\t"<<object2.*pd<<"\n";
 cout<<"function : ";
 (object1.*pf)();
 cout<<"\t";
 (object2.*pf)();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Values object : 4      5
function : T = 4      T = 5
Press any key to continue . . .
```

```
// S09-22.cpp
// تخصیص پویای حافظه به شی

#include "stdafx.h"
#include "iostream"

using namespace std;

class stud
{
 int id;
 char name[50];
 char family[50];
public:
 void input(void);
 void print(void);
};

void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}

void stud::print(void)
{
 cout<<"\nId = "<<id;
 cout<<"\nName = "<<name;
 cout<<"\nFamily = "<<family;
}

int main(void)
{
 stud *s;
 s=new stud;
 if(!s)
 {
  cout<<"Allocation failure.";
  exit(0);
 }
 s->input();
 s->print();
 delete s;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter Id,Name,Family : 100 Hooman Sayyari

Id = 100
Name = Hooman
Family = Sayyari
Press any key to continue . . .
```

```
// S09-23.cpp
// ایجاد آرایه ای از اشیا به صورت پویا

#include "stdafx.h"
#include "iostream"

using namespace std;

class stud
{
 int id;
 char name[50];
 char family[50];
 public:
  void input(void);
  void print(void);
};

void stud::input(void)
{
 cout<<"Enter Id,Name,Family : ";
 cin>>id>>name>>family;
}

void stud::print(void)
{
 cout<<"\nId = "<<id;
 cout<<"\nName = "<<name;
 cout<<"\nFamily = "<<family;
}

int main(void)
{
 int i;
 stud *s;
 s=new stud[5];
 if(!s)
 {
  cout<<"Allocation failure.";
  exit(0);
 }
 for(i=0;i<5;i++)
  s[i].input();
 for(i=0;i<5;i++)
  s[i].print();
 delete [] s;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Enter Id,Name,Family : 101 Hooman Sayyari
Enter Id,Name,Family : 102 Ali Bahmani
Enter Id,Name,Family : 103 Reza Soleimani
Enter Id,Name,Family : 104 Pedram Ahmadi
Enter Id,Name,Family : 105 Dariush Dehroyeh

Id = 101
Name = Hooman
Family = Sayyari
Id = 102
```

```
Name = Ali
Family = Bahmani
Id = 103
Name = Reza
Family = Soleimani
Id = 104
Name = Pedram
Family = Ahmadi
Id = 105
Name = Dariush
Family = Dehroyeh
Press any key to continue . . .
```

```
// P09-01.cpp
// تـابـع دوست //

#include "stdafx.h"
#include "iostream"

using namespace std;

class myclass
{
 int num;
public:
 myclass(int x)
 {
  num=x;
 }
 friend int isneg(myclass ob);
};

int isneg(myclass ob)
{
 return(ob.num<0)?1:0;
}

int main(void)
{
 myclass a(-1),b(2);
 cout<<isneg(a)<<"   "<<isneg(b);
 cout<<endl;
 system("pause");
 return 0;
}
```

```
1 0
Press any key to continue . . .
```

```
// P09-02.cpp
// هماهنگی بین ۲ کلاس برای دسترسی به چاپگر با استفاده از تابع دوست

#include "stdafx.h"
#include "iostream"

using namespace std;

class pr2;

class pr1
{
 int printing;
public:
 pr1(void)
 {
  printing=0;
 }
 void set_print(int status)
 {
  printing=status;
 }
 friend int inuse(pr1 object1,pr2 object2);
};

class pr2
{
 int printing;
public:
 pr2(void)
 {
  printing=0;
 }
 void set_print(int status)
 {
  printing=status;
 }
 friend int inuse(pr1 object1,pr2 object2);
};

int inuse(pr1 object1,pr2 object2)
{
 if(object1.printing||object2.printing)
   return 1;
 else
   return 0;
}

int main(void)
{
 pr1 p1;
 pr2 p2;
 if(!inuse(p1,p2))
  cout<<"Printer idle\n";
 cout<<"Setting p1 to printing...\n";
 p1.set_print(1);
 if(inuse(p1,p2))
  cout<<"Now, printer in use.\n";
 cout<<"Turn off p1...\n";
 p1.set_print(0);
 if(!inuse(p1,p2))
  cout<<"Printer idle\n";
 cout<<"Turn on p2...\n";
 p2.set_print(1);
```

```
 if(inuse(p1,p2))
  cout<<"Now, printer in use.\n";
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Printer idle
Setting p1 to printing...
Now, printer in use.
Turn off p1...
Printer idle
Turn on p2...
Now, printer in use.
Press any key to continue . . .
```

```
// P09-03.cpp
// آرایه ای از اشیا

#include "stdafx.h"
#include "iostream"

using namespace std;

class MyClass
{
 int h;
 int i;
public:
 MyClass(int j,int k)
 {
  h=j;
  i=k;
 }
 int getInt(void)
 {
  return i;
 }
 int getHeight(void)
 {
  return h;
 }
};

int main(void)
{
 MyClass myObject[3]={
  MyClass(1,2),
  MyClass(3,4),
  MyClass(5,6)
 };
 int i;
 for(i=0;i<3;i++)
 {
  cout<<myObject[i].getHeight();
  cout<< ",";
  cout<<myObject[i].getInt()<<"\n";
 }
 system("pause");
 return 0;
}
```

```
1,2
3,4
5,6
Press any key to continue . . .
```

فصل نهم: کلاس مشتق

```
// S13-01.cpp
// (Derived) کلاس مشتق

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
int x;
public:
 void input(int a)
 {
  x=a;
 }
 void print(void)
 {
  cout<<"\nx="<<x;
 }
};
class DerivedClass:public BaseClass
{
 int y;
public:
 DerivedClass(int a)
 {
  y=a;
 }
 void show(void)
 {
  cout<<"\ny="<<y;
 }
};
int main(void)
{
 DerivedClass d(5);
 d.input(6);
 d.print();
 d.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x=6
y=5
Press any key to continue . . .
```

```
// S13-02.cpp
// ترتیب اجرای سازنده و مخرب کلاس پایه و مشتق

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
public:
 BaseClass(void)
 {
   cout<<"Constructing BaseClass.\n";
 }
 ~BaseClass(void)
 {
   cout<<"Destructing BaseClass.\n";
 }
};

class DerivedClass:public BaseClass
{
public:
 DerivedClass(void)
 {
   cout<<"Constructing DerivedClass.\n";
 }
 ~DerivedClass(void)
 {
  cout<<"Destructing DerivedClass.\n";
 }
};

int main(void)
{
 DerivedClass d;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Constructing BaseClass.
Constructing DerivedClass.

Press any key to continue . .
Destructing DerivedClass.
Destructing BaseClass.
```

```
// S13-03.cpp
// دسترسی یک شی از کلاس مشتق به عضو محافظت شده کلاس پایه

#include "stdafx.h"
#include "iostream"

using namespace std;

class twonumbers
{
protected:
 int x,y;
public:
 void input(void)
 {
  cout<<"\nEnter two Numbers : ";
  cin>>x>>y;
 }
 void print(void)
 {
  cout<<"\nx = "<<x<<"\ty = "<<y;
 }
};

class op:public twonumbers
{
public:
 void sum(void)
 {
  cout<<"\nsum = "<<x+y;
 }
 void mult(void)
 {
  cout<<"\nmult = "<<x*y;
 }
};

int main(void)
{
 op d;
 d.input();
 d.print();
 d.sum();
 d.mult();
 cout<<endl;
 system("pause");
 return 0;
}
```

Enter two Numbers : 5 10

x = 5 y = 10
sum = 15
mult = 50
Press any key to continue . . .

```
// S13-04.cpp
// ارسال مقدار سازنده کلاس پایه توسط سازنده کلاس مشتق

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
 int x;
public:
 BaseClass(int a)
 {
  x=a;
 }
 void print(void)
 {
  cout<<"\nx="<<x;
 }
};

class DerivedClass:public BaseClass
{
 int y;
public:
 DerivedClass(int a,int b):BaseClass(b)
 {
  y=a;
 }
 void show(void)
 {
  cout<<"\ny="<<y;
 }
};

int main(void)
{
 DerivedClass d(4,6);
 d.print();
 d.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x=6
y=4
Press any key to continue . . .
```

```
// S13-05.cpp
// دسترسی به کلاس پایه و کلاس مشتق توسط اشاره گر

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
 int x;
public:
 BaseClass(int a)
 {
  x=a;
 }
void print(void)
 {
 cout<<"\nx="<<x;
 }
};

class DerivedClass:public BaseClass
{
 int y;
public:
 DerivedClass(int a,int b):BaseClass(b)
 {
  y=a;
 }
 void show(void)
 {
  cout<<"\ny="<<y;
 }
};

int main(void)
{
 BaseClass b(6),*bp;
 DerivedClass d(7,5);
 bp=&b;
 bp->print();
 bp=&d;
 bp->print();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x=6
x=5
Press any key to continue . . .
```

```
// S13-06.cpp
// تغییر عملکرد تابع کلاس پایه در کلاس مشتق

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
public:
 void show(void)
 {
  cout<<"this class is base.";
 }
};

class DerivedClass:public BaseClass
{
public:
 void show(void)
 {
  cout<< "This class is derived.";
 }
};

int main(void)
{
 BaseClass b;
 cout<<"\nb.show : ";
 b.show();
 DerivedClass d;
 cout<<"\nd.show : ";
 d.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
b.show : this class is base.
d.show : This class is derived.
Press any key to continue . . .
```

```
// S13-07.cpp
// تغییر عملکرد تابع کلاس پایه در کلاس مشتق و استفاده از تابع پایه در کلاس مشتق

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass
{
public:
 void show(void)
 {
  cout<<"This class is base.\n";
 }
};

class DerivedClass:public BaseClass
{
public:
 void show(void)
 {
  BaseClass::show();
  cout<<"This class is derived.\n";
 }
};

int main(void)
{
 DerivedClass d;
 cout<<"DerivedClass.show : \n";
 d.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
DerivedClass.show :
This class is base.
This class is derived.
Press any key to continue . . .
```

```
// S13-08.cpp
// وراثت چند گانه

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass1
{
protected:
 int x;
public:
 BaseClass1(int a)
 {
  x=a;
 }
};

class BaseClass2
{
protected:
 int y;
public:
 BaseClass2(int b)
 {
  y=b;
 }
};

class DerivedClass:public BaseClass1,public BaseClass2
{
public:
 DerivedClass(int a,int b):BaseClass1(a),BaseClass2(b)
 {
 }
 void show(void)
 {
  cout<<" x = "<<x<<"  "<<" y = "<<y;
 }
};

int main(void)
{
 DerivedClass d(4,6);
 d.show();
 cout<<endl;
 system("pause");
 return 0;
}
```

```
x = 4  y = 6
Press any key to continue . . .
```

```
// S13-09.cpp
// اعضای مشترک کلاس های پایه در وراثت چند گانه

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass1
{
public:
 int x;
 BaseClass1(int a)
 {
  x=a;
 }
};

class BaseClass2
{
public:
 int x;
 BaseClass2(int b)
 {
  x=b;
 }
};

class DerivedClass:public BaseClass1,public BaseClass2
{
public:
 DerivedClass(int a,int b):BaseClass1(a),BaseClass2(b)
 {
 }
};

int main(void)
{
 DerivedClass d(4,6);
 cout<<"\nd.BaseClass1::x = "<<(d.BaseClass1::x);
 cout<<"\nd.BaseClass2::x = "<<(d.BaseClass2::x);
 cout<<endl;
 system("pause");
 return 0;
}
```

d.BaseClass1::x = 4
d.BaseClass2::x = 6
Press any key to continue . . .

```
// P13-01.cpp
// سازنده و مخرب در وراثت چند گانه

#include "stdafx.h"
#include "iostream"

using namespace std;

class BaseClass1
{
 public:
 BaseClass1()
 {
  cout<<"constructing BaseClass1.\n";
 }
 ~BaseClass1()
 {
  cout<<"Constructing BaseClass1.\n";
 }
};

class BaseClass2
{
public:
 BaseClass2()
 {
  cout<<"Constructing BaseClass2.\n";
 }
 ~BaseClass2()
 {
  cout<<"Constructing BaseClass2.\n";
 }
};

class DerivedClass:public BaseClass1,public BaseClass2
{
public:
 DerivedClass()
 {
  cout<<"Constructing DerivedClass.\n";
 }
 ~DerivedClass()
 {
 cout<<"Destructing DerivedClass.\n";
 }
};

int main(void)
{
 DerivedClass d;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
constructing BaseClass1.
Constructing BaseClass2.
Constructing DerivedClass.

Destructing DerivedClass.
Constructing BaseClass2.
Constructing BaseClass1.
Press any key to continue . . .
```

```
// P13-02.cpp
// محاسبه مساحت اشکال مختلف با چند کلاس

#include "stdafx.h"
#include "iostream"

using namespace std;

class AreaClass
{
public:
 double height;
 double width;
};

class Rectangle:public AreaClass
{
public:
 Rectangle(double h,double w)
 {
  height=h;
  width=w;
 }
 double area()
 {
  return height*width;
 }
};

class Isosceles:public AreaClass
{
public:
 Isosceles(double h,double w)
 {
  height=h;
  width=w;
 }
 double area()
 {
  return 0.5*width*height;
 }
};

class cylinder:public AreaClass
{
public:
 cylinder(double h,double w)
 {
  height=h;
  width=w;
 }
 double area()
 {
  return(2*3.14*(width/2)*(width/2))+(3.14*width*height);
 }
};

int main(void)
{
 Rectangle rectangleObject(10.0 , 5.0);
 Isosceles triangleObject(4.0 , 6.0);
 cylinder cylinderObject(3.0 , 4.0);
 cout<<"Rectangle:"<<rectangleObject.area()<<endl;
 cout<<"Triangle:"<<triangleObject.area()<<endl;
```

```
 cout<<"Cylinder:"<<cylinderObject.area()<<endl;
 cout<<endl;
 system("pause");
 return 0;
}
```

```
Rectangle:50
Triangle:12
Cylinder:62.8
Press any key to continue . . .
```

```
// P13-03.cpp
// کلاس پایه برای گرفتن و چاپ رشته و کلاس مشتق برای محاسبه طول رشته

#include "stdafx.h"
#include "iostream"

using namespace std;

class mybase
{
 char str[80];
public:
 mybase(char *s)
 {
  strcpy(str,s);
 }
 char *get()
 {
  return str;
 }
};

class myderived:public mybase
{
 int len;
public:
 myderived(char *s):mybase(s)
 {
  len=strlen(s);
 }
 int getlen()
 {
  return len;
 }
 void show()
 {
  cout<<get()<<"\n";
 }
};

int main(void)
{
 myderived ob("hello");
 ob.show();
 cout<<ob.getlen()<<"\n";
 cout<<endl;
 system("pause");
 return 0;
}
```

```
hello
5
Press any key to continue . . .
```