مفهوم کلاسها

اصولا برنامه سازی شی گرا این امکان را به برنامه نویس می دهد تا اشیای دنیای واقعی را با استفاده از اشیای متناظر در کد به طور دقیق پیاده سازی کند. در واقع به صورت نظری اگر بتوان شی را تعریف کرد پس می توان برنامه آن را نیز نوشت،حتی اگر این شی انسان باشد!(در واقع چون تعریفی دقیق از انسان وجود ندارد بنابراین نمی توان یک شی از ان را تعریف کرد)
هرشی (چه در دنیای واقعی و چه در دنیای برنامه نویسی شی گرای) دارای خصوصیاتی است و می تواند کارهایی انجام دهد. این ابتدایی ترین تعریف از یک شی می باشد .
به عنوان مثال یک اتومبیل دارای خصوصیاتی نظیر :تعداد چرخ ها و تعداد سیلندرهاو شماره سیلندرو سرعت و وزن و غیره ... می باشد.و می تواند کارهایی از قبیل حرکت کردن و کم کردن سرعت و سرعت گرفتنو غیره را انجام داد .
این دودید در زبانهای رویه ای مانند c در هنگام پیاده سازی کاملا" جدا از هم در نظر گرفته می شوند:یعنی متغیر هایی برای خصوصیات و توابعی برای کارهایی که می تواند انجام گیرد .ولی در زبانهای شی گرا مانند c++ این امکان وجود دارد که خصوصیات (داده ها)و کارها(رویه ها) در یک مفهوم واحد بنام شی کپسوله شود و با یک نام واحد شناخته شده و به آن دسترسی شود .
در زبانهای رویه ای به طور دقیق مشخص نیست که چه توابعی روی چه داده هایی عمل می کنند. ولی در زبانهای شی گرا مانند c++ با توجه به استفاده از کپسوله سازی و این که هر شی دارای خصوصیاتی (داده هایی) است که فقط رویه های مربوط به آن شی می تواند به آن دسترسی داشته باشد دقیقا مشخص است که چه توابعی روی یک شی و یا داده هایی ان شی کار می کنند .
در زبان c++ برای پیاده سازی یک شی از کلاس ها استفاده می شود.در واقع از کلاسها برای تعریف یک نوع جدید استفاده می شود.در این تعریف کلیه خصوصیت و عملکرد های این نوع تعریف می شوند. پس از تعریف به راحتی می توان هر تعداد شی که لازم باشد از این نوع درست کرد; اشیا از یک نوع و دارای خصوصیات)متغیر ها)و عملکرد هایی (توابعی) یکسان می باشند.به عنوان مثال نوعی به نام دانشجو فرض می شودکه دارای خصیصه های شماره دانشجویی ،نام ونام خانوادگی می باشد و دو دانشجو(شی) وجود دارد; با اینکه هر دو این خصوصیات را خواند داشت ولی الزاما دارای یک مقدار نیستند

نحوه تعریف یک کلاس

کلاس هارا با استفاده از کلید واژه class تعریف می کنند،شکل کلی تعریف یک کلاس به صورت زیر می باشد

Class class-name {

//data member and member function
};
سمی کالون بعد از براکت بسته این نکته را نشان می دهد که این مجموعه کد،تعریف یک نوع جدید را موجب می شود.همانطور که در تعریف ساختمان در زبان c نیز از سمی کالون در آخر تعریف استفاده می شود.برنامه نویسان مبتدی معمولا در در هنگام کد نویسی این سمی کالن اتمام کننده تعریف را فراموش می کنند که باعث تولید خطای زمان کامپایل می شود. دقت کنید که در هنگام تعریف کلاس سمی کولن را فراموش نکنید

کلاس

کلاس روشی برای بسته بندی نوع داده مجرد است . در کلاس امکان بسته بندی و محصور کردن (Encapsulation)مجموعه ای از داده ها است . روال های پردازش کننده این داده ها را به صورت یک بسته فراهم می کند.

داده های داخل یک کلاس به وسیله کلاس محافظت می گردد. به گونه ای که پردازش داده های خصوصی یک کلاس از طریق روال های دخلی آن امکان پذیر است . داده های یک کلاس را متغیرهای کلاس و روال های آن را روش نامیده اند.

برای مثال کلاس انسان ها یک کلاس قابل تعریف است. در این کلاس خصوصیات مشترک انسانها تعریف می گردد و هیچ انسان خاصی را نشان نمی دهد. کلاس یک نوع است. همانگونه که مثلا int یک نوع است.

عملیات محاسباتی (یا غیر محاسباتی) بر روی نوع داده انجام نمی شود. بلکه این عملیات بر روی متغیرهایی که از این نوع داده تعریف می گردد انجام می شود. به طور مشابه عملیات محاسباتی (یا غیر محاسباتی) روی کلاس انجام نمی شود

در تعریف بالا class-name نامی است که به این کلاس اطلاق می شود،ودر ادامه برنامه این کلاس با این نام شنبخته خواهد شد،در واقع نامی است برای یک نوع جدید .از نوع های پیش تعریف شده می توان به int,float و غیره اشاره کرد.بعد در داخل براکت ها({})اعضای این کلاس تعریف می شوند که شامل تمامی داده ها .رویه های مربوط به این کلاس می باشد.
به خصیصه ها و یا همان داده های داخل یک کلاس عضو داده ای گفته می شودو رویه های موجود در تعریف یک کلاس را توابع عضو می نامند.
متالی:
کلاس زیر یک اتومبیل را تعریف می کند،اتومبیل دارای خصوصیات تعداد درها و تعداد چرخ ها می باشد ومیتوان سرعت خود را افزایش دهدو یا متوقف شود.

```
Class ccar

{
   Unsigned int numberofdoors;
   Unsigned int numberofwheels;
   Void speedup()'
   Void stop();
};
```

به طور پیش فرض اعضای کلاس بهصورت خصوصی می باشند واز خارج از کلس قابل دسترسی نیستند.یعنی در تعریف بالا خصوصیات number of doorsو number of wheels و توابع speedup()و stop() حالت خصوصی دارندواز خارج از کلاس قابل دسترسی نیستند،برای مهیا سازی امکان دسترسی به توابع از خارج از کلاس آن ها را باید به صورت عمومی تعریف کرد که این کار با استفاده از تصریح کننده سطح دسترسی public انجام می گیرد.
حتی در دنیای واقعی نیز بیشتر اشیا عملکرد هایی به صورت عمومی دارند که امکان ارتباط از بیرون را برای آن اشیا مهیا می سازد که مثلا" اتومبیل از طریق پدال هاو فرمان و دنده ها با راننده در ارتباط است.و حالت عمومی دارند ولی کارهایی از قبیل میزان تزریق بنزین به کاربراتور فاشتغال بنزین در سیلندر،توابعی برای انجام اعمال گیربکس و غیره به صورت خصوصی بوده و نیازی به عمومی بودن آنها نیست.

تصریح کننده های سطح دسترسی

++Cدارای سه تصریح کننده سطح دسترسی برای اعضای کلاس می باشندکه برای تعیین محدوده دسترسی اعضای کلاس از خارج از کلاس به کار می روند.این سه تصریح کننده در جدول زیر آورده شده اند:

PRIVATE(خصوصی):متغیر یا تابعی که این خصوصیات را دارا باشد نمی تواند خارج از کلاس مربوط مورد دستیابی قرار گیرد.

PUBLIC : اعضای عمومی می توانند توسط هر بخشی از خارج از کلاس مورد دستیابی قرار گیرند .

PROTECTED : اعضای حفاظت شده نیز مانند اعضای خصوصی نمی توانند خارج از کلاس مورد دسترسی قرار گیرند.تفاوت این تصریح کننده با PRIVATE در هنگام ارث بری نمایان می شودکهدر این باره توضیح خواهیم داد.

ما در اینجا از PUBLIC,PRIVATE استفاده خواهد شد تا معین شود که یک عضو به خصوص از کلاس

از خارج کلاس قابل دسترسی می باشد یا نه.
همانطور کهگفتم به طور پیش فرض سطح دسترسی اعضای کلاس به صورت PRIVATE می باشد.برای تعریف اعضایی که خارج از کلاس نیز قابل استفاده باشنداز کلید واژه PUBLIC به همراه ":" استفاده می شود.بعد از این کلید واژه کلیه اعضایی که تعریف می شوند با سطح دسترسی عمومی شناخته خواهند شدو از خارج از کلاس قابل دسترسی خواهند بود.
شکل کلی استفاده از ترصریح کننده ها به صورت زیر می باشد :

```
Class class-name

{
// private members
Public:
// public members
};
```

مثال:کلاس اتومبیل نوشته شده در مثال قبل را به صورت زیر تصحیح می کنیم:

```
Class Ccar
{
Unsigned int numberofdoors;
Unsigned int numberofwheels;
Unsigned int speed;
Public:
Void speedup()'
Void stop();
};
```

توضیح:
چون متغیر های numberofdoorsو numberofwheels و;speed بلافاصله بعد از آکولاد باز تعریف کلاس آمده اند به طور پیش فرض دارای سطح دسترسی PRIVATE می باشند و از خارج از کلاس قابل دسترسی نمی باشند. ولی توابع ()SPEEDUP و ()STOPچون به صورت PUBLIC معرفی شده اند،پس در خارج کلاس نیز می توانند فراخوانی شوند.

همانطور که دیده می شود،در کلاس Ccar توابع ()SPEEDDUPو ()STOP معرفی شده اند ولی هنوز تعریف نشده اند. به این نوع تعریف کلاس که در آن درباره خصوصیات شی گرا و کارهایی که می تواند انجام دهدصحبت شده ولی درباره نحوه پیاده ساری ان ADT گفته می شود.

نحوه تعریف توابع عضو

بعد از تعریف کلاس و معرفی توابع عضو آن کلاس باید توابع عضو کلاس تعریف شوند و نحوه پیاده سازی آنها معین شود.در زبان c++ نحوه تعریف یک تابع عضو همانند یک تابع معمولی می باشدبا این تفاوت که عضو یک کلاس می باشدو در هنگام تعریف این تابع با ید کلاسی که در آن عضو می باشد نیز ذکر شود تا برای کامپایلر مشخص شودکه مقصود تعریف تابعی استکه در داخل یک کلاس مشخص قرار دارد.در هنگام تعریف یک تابع عضو از عملگرد :: استفاده می شود. شکل کلی تعریف توابع عضو به صورت زیر می باشد:

()نام تابع:: نام کلاس نوع خروجی
{
بئنه تابع عضو در اینجا قرار می گیرد
}

به عملکرد "::"عملگر تعیین میدان دیدگفته می شود
به عنوان مثال برای تعریف تابع speedup که عضو کلاس Ccar است به صورت زی کد نویسی می شود:

```
Void Ccar ::speedup()
{
//…..
}
```

نکاتی چند در مورد تعریف کلاس

ذکر تصریح کننده Private ضروری نیست با این همه استفاده از آن و تعریف صریح آن توصیه می

شود ود سراسر کتاب به طور صریح استفاده خواهد شد.

نکته:از دیدگاه برنامه نویسی شی گرا بهتر است همه متغیرهای عضو یک کلاس به صورت

PRIVATE و یا PROTECTED تعریف شوند.این کار باعث می شود تا احتمال دستکاری مستقیم مقادیر این متغیر ها از خارج کلاس از بین برودو مفهوم کپسوله سازی به طور کامل تری رعایت شود.همچنین بهتر است تنها توابع عضوی به صورت PUBLIC تعریف شوند که نقش ارتباطی کلاس و دنیای خارج را بر عهده دارند.

نقش ارتباطی در اشیای موجود در طبیعت نیز وجود داردمثلا یک انسان می تواند به طور ارادی نفس بکشد که به نوبه خود باعث جریان یافتن هوا در ریه ها و فعل و انفعالات مویژه ها در رویه و حرکت اکسیژن در رگها می شود; ولی یک انسان به طور ارادی نمی تواند هیچ کنترلی روی مویژه های ریه یا میزان اکسیزن انتقالی توسط رگ ها داشته باشد.

نحوه تعریف یک شی

تعریف کلاس یک مفهوم انتزاعی می باشد یعنی با تعریف کلاس تنها یک نوع جدید تعریف می شودوهیچ متغیری تعریف نمی شودوحافظه ای به آن تخصیص نیافته است.کلاس تنها تعریف یک نوع جدید می باشد که می توان اشیایی از این نوع جدید تعریف کرد.

همان طور که متغیر هایی از نوع های درون ساخته c++ مانند int, char می توان ساخت برای تعریف یک متغیر از نوع یک کلاس که به آن شی یا نمونه گفته می شوداز شکل کلی زیر استفاده می شود:

```
Class_name object_name;
```

Class_nameنام کلاسی می باشد که می خواهیم یک شی به نام object_name از ان نوع تعریف کنیم .به عنوان مثال برای تعریف شی با نام Elegance از نوع کلاس Ccar از کدی به صورت

زیر می توان استفاده کرد
Ccar elegance;

مثال
برنامه زیر بروز خطا هنگام تلاش برای دسترسی به اعضای خصوصی کلاس را نشان می دهد

```
#include<iostrem.h>
#include<conio.h>

//----------------------------------------------Ccar class
Class Ccar
{
Private:

Unsigned int numberofdoors;

Unsigned int numberofwheels;

Undigned int speed;
};
//----------------------------------------------main function
Int main()
{
Ccar elegance;
//…
Elegance.speed=100; //error

//…
Return 0;
}
```

توضیح برنامه

برنامه زیر موجب بروز پیغام خطا شده و کامپایل نمی شود در خط 20 این برنامه ،از داخل تابع main سعی شده است تا به عضو خصوصی کلاس Ccar دسترسی پیدا شود:

Is not accessible!elegance.speed=100;//error'Ccar::speed'

این کار ممکن نیست زیرا همانطور که ذکر شد تنها توابع عضو کلاس می توانند به اعضای خصوصی دسترسی پیدا کنند و اعضای خصوصی به طور کل از دید خارج از کلاس مخفی می باشند.از طرف دیگر از عملگر "." تنها برای دسترسی به اعضای عمومی کلاس می توان استفاده کرد.