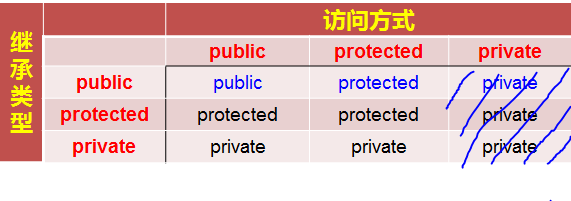
1. 编写程序，定义具有继承关系的圆circle和带边框的圆bordercircle，圆的数据成员包括半径r，函数成员包含：构造函数、析构函数、输入半径input、获取半径、面积以及周长。带边框的圆的数据成员包含边框宽度w，函数成员包含：输入边框宽度input、内圆面积、边框面积。在主函数中用带边框的圆对所有函数进行测试。

（提示：两个input函数为同名函数）

1. 下面代码中的继承方式分别用Public、Protected、Private，访问权限有何变化？



#include<iostream>

using namespace std;

class A

{

public:

int x;

A()

{

x=10;

y=20;

z=30;

}

protected:

int y;

private:

int z;

};

class B:public A

{

public:

void fun()

{

cout<<x<<endl;

cout<<y<<endl;

// cout<<z<<endl;

}

};

int main()

{

B b;

cout<<b.x<<endl;

cout<<b.y<<endl;

// cout<<b.z<<endl;

}

1. 编写程序，定义具有继承关系的点和圆类，点的数据成员包括X、Y坐标，函数成员包含SETX，SETY，SHOWXY。圆的数据成员包含半径R，函数成员包含SETR、SHOW及AREA。

【参考代码】

1、#include<iostream>

using namespace std;

const double pi=3.14;

class circle

{

private:

double r;

public:

double area()

{

return pi\*r\*r;

}

double len()

{

return 2\*pi\*r;

}

void input()

{

cin>>r;

}

double getr()

{

return r;

}

circle()

{

r=0;

}

circle(double \_r)

{

r=\_r;

}

~circle()

{}

};

class bordercircle:public circle

{

private:

double w;

public:

double innerarea()

{

double x;

x=circle::getr();

return pi\*(x-w)\*(x-w);

}

double borderarea()

{

return circle::area()-innerarea();

}

void input()

{

circle::input();

cin>>w;

}

};

int main()

{

bordercircle obj;

obj.input();

cout<<obj.len()<<endl;

cout<<obj.innerarea()<<endl;

cout<<obj.borderarea ()<<endl;

}