While(true)

while(true)是一个无穷循环语句  我们必须在他的循环语句内部加入一个判断  当他达到了什么要求就会跳出

int i =1

while(true)

{undefined

if(i>10)break;

i++;

}这样最后弹出的就是11；

在类定义时定义成员函数  
        成员函数的实现可以在类定义时同时完成, 如代码:

1. #include <iostream>
3. using namespace std;
5. class Point
7. {
9. public:
11. void setPoint(int x, int y) *//实现setPoint函数*
13. {
15. xPos = x;
17. yPos = y;
19. }

22. void printPoint() *//实现printPoint函数*
24. {
26. cout<< "x = " << xPos << endl;
28. cout<< "y = " << yPos << endl;
30. }

33. private:
35. int xPos;
37. int yPos;
39. };

42. int main()
44. {
46. Point M; *//用定义好的类创建一个对象 点M*
48. M.setPoint(10, 20); *//设置 M点 的x,y值*
50. M.printPoint(); *//输出 M点 的信息*

53. return 0;
55. }

运行输出:

1. x = 10
2. y = 20

与类的定义相比, 在类内实现成员函数不再是在类内进行声明, 而是直接将函数进行定义, 在类中定义成员函数时, 编译器默认会争取将其定义为 inline 型函数 2>. 在类外定义成员函数  
        在类外定义成员函数通过在类内进行声明, 然后在类外通过作用域操作符 **::** 进行实现, 形式如下:

1. 返回类型 类名::成员函数名(参数列表)
2. {
3. *//函数体*
4. }


8. #include <iostream>
10. using namespace std;
12. class Point
14. {
16. public:
18. void setPoint(int x, int y); *//在类内对成员函数进行声明*
20. void printPoint();

23. private:
25. int xPos;
27. int yPos;
29. };
31. void Point::setPoint(int x, int y) *//通过作用域操作符 '::' 实现setPoint函数*
33. {
35. xPos = x;
37. yPos = y;
39. }
41. void Point::printPoint() *//实现printPoint函数*
43. {
45. cout<< "x = " << xPos << endl;
47. cout<< "y = " << yPos << endl;
49. }
51. int main()
53. {
55. Point M; *//用定义好的类创建一个对象 点M*
57. M.setPoint(10, 20); *//设置 M点 的x,y值*
59. M.printPoint(); *//输出 M点 的信息*

62. return 0;
64. }

依 setPoint 成员函数来说, 在类内声明的形式为 **void setPoint(int x, int y);** 那么在类外对其定义时函数头就应该是 **void Point::setPoint(int x, int y)** 这种形式, 其返回类型、成员函数名、参数列表都要与类内声明的形式一致。

if(P)的含义：

指针判断都要跟NULL比较，相当于0，假值，其余都是真值

if(P)就是if(P != NULL ), 不为空执行语句

if(！P)就是if(P == NULL ), 为空执行语句