```
#include "pch.h"
#include "CAppObject.h"

// Public constructors
CAppObject::CAppObject(CRect bounds, CString ID, BOOL active)
{
    this->bounds = bounds;
    this->ID = ID;
    this->active = active;
}
CAppObject::~CAppObject()
{
}

void CAppObject::draw(CDC* pDc)
{
}

// Public message handlers
void CAppObject::OnSize(UINT nType, int cx, int cy)
{
}

BOOL CAppObject::OnLButtonUp(UINT nFlags, CPoint point)
{
    return 0;
}
BOOL CAppObject::OnLButtonDown(UINT nFlags, CPoint point)
{
    return 0;
}
void CAppObject::OnLButtonDblClk(UINT nFlags, CPoint point)
{
}
void CAppObject::OnRButtonUp(UINT nFlags, CPoint point)
{
}
void CAppObject::OnRButtonDown(UINT nFlags, CPoint point)
{
}
BOOL CAppObject::OnMouseMove(UINT nFlags, CPoint point)
{
    return TRUE;
}
int CAppObject::OnRecieveText(CString text)
{
    return 0;
}
void CAppObject::OnRecieveBackspace()
{
}
void CAppObject::OnRecieveReturn()
{
}
void CAppObject::OnRecieveTab()
{
}

void CAppObject::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

BOOL CAppObject::OnContextMenu(CWnd* pWnd, CPoint point)
{
    return 0;
}

// Public getters, setters and checkers
CString CAppObject::getID()
{
    return this->ID;
}
void CAppObject::setID(CString ID)
{
    this->ID = ID;
}

CRect CAppObject::getBounds()
{
    return this->bounds;
}
BOOL CAppObject::checkPointInBounds(CPoint point)
{
    if (this->bounds.left <= point.x && bounds.right >= point.x &&
        this->bounds.top <= point.y && this->bounds.bottom >= point.y) {
        return TRUE;
    }

    return FALSE;
}

void CAppObject::setBounds(CRect bounds)
{
    this->bounds = bounds;
}
void CAppObject::setBounds(int x, int y, int cx, int cy)
{
    this->bounds.left = x;
    this->bounds.top = y;
    this->bounds.right = x + cx;
    this->bounds.bottom = y + cy;
}
void CAppObject::setPosition(CPoint point)
{
    this->move(point.x - this->bounds.left, point.y - this->bounds.top);
}
void CAppObject::move(int x, int y)
{
    this->bounds.left += x;
    this->bounds.right += x;
    this->bounds.top += y;
    this->bounds.bottom += y;
}

BOOL CAppObject::isActive()
{
    return this->active;
}
void CAppObject::setActive(BOOL val)
{
    this->active = val;
}

```