```
// Wyo C++ Ch. 10 Demo #3

#include <iostream.h>
#include "H:\C++\ClassFiles\apstring.h"

struct Tank
{
    // member variables
    apstring color;
    int x;
    int y;
    int ammo;
    char direction;

    // member functions
    void move(int);
    void turn(int);

    // constructors
    Tank():color("BLUE"), x(0), y(0), ammo(10), direction('W'){}    // default constructor

                        // an initializer list (i.e. the stuff after the colon)
                        // is being used with the default constructor above

    Tank(apstring, int, int, int, char);
    Tank(apstring, int, int);
};

int main()
{
    Tank playerA;
    Tank playerB("BLUE", 100, 100, 10, 'N');

    apstring userColor;
    int userX = 0;
    int userY = 0;

    cout << "Enter your tank's color, initial X position, & initial Y position: ";
    cin >> userColor >> userX >> userY;

    Tank playerC(userColor, userX, userY);

    cout << "PlayerA is located at: " << playerA.x << ", " << playerA.y << endl;
    cout << "PlayerB is located at: " << playerB.x << ", " << playerB.y << endl;
    cout << "PlayerC is located at: " << playerC.x << ", " << playerC.y << endl;

    return 0;
}// end of main

//Tank::Tank():color("BLUE"), x(0), y(0), ammo(10), direction('W')
//{
//    color = "BLUE";   ********** default constructor function definition is
//    x = 0;            ********** unnecessary since an initializer list was
//    y = 0;            ********** used with the function prototype in Tank's
//    ammo = 10;        ********** struct definition
//    direction = 'W';
//}

Tank::Tank(apstring myColor, int myX, int myY, int myAmmo, char myDirxn)
:color(myColor), x(myX), y(myY), ammo(myAmmo), direction(myDirxn)
{
    //color = myColor;  ********** since an initializer list is used with the
    //x = myX;          ********** function header, the assignment statements
    //y = myY;          ********** in the body aren't necessary
    //ammo = myAmmo;
    //direction = myDirxn;
}

Tank::Tank(apstring myColor, int myX, int myY)
{
    color = myColor;
    x = myX;
    y = myY;
    ammo = 10;
    direction = 'N';
}
```

```
void Tank::move(int moveAmount)
{
    switch (direction)
    {
    case 'N':
        y -= moveAmount;
        break;
    case 'S':
        y += moveAmount;
        break;
    case 'W':
        x -= moveAmount;
        break;
    case 'E':
        x += moveAmount;
        break;
    }

}

void Tank::turn(int turnAmount)
{
    turnAmount = turnAmount % 4;

    switch (direction)
    {
    case 'N':

        if      (turnAmount == 1)   direction = 'E';
        else if (turnAmount == 2)   direction = 'S';
        else if (turnAmount == 3)   direction = 'W';

        break;
    case 'E':

        if      (turnAmount == 1)   direction = 'S';
        else if (turnAmount == 2)   direction = 'W';
        else if (turnAmount == 3)   direction = 'N';

        break;
    case 'S':

        if      (turnAmount == 1)   direction = 'W';
        else if (turnAmount == 2)   direction = 'N';
        else if (turnAmount == 3)   direction = 'E';

        break;
    case 'W':

        if      (turnAmount == 1)   direction = 'N';
        else if (turnAmount == 2)   direction = 'E';
        else if (turnAmount == 3)   direction = 'S';

        break;
    }

}
```