```cpp
/*
 * Class name:  TradeItem (header file)
 * Class description:  Class definition for the type TradeItem.  TradeItem objects represent an item that can be sold or
 *      traded.  Each instance has a name, a value, and a flag indicating whether the item is legal or illegal to have.
 *
 * Programmer:  Chad Philip Johnson
 * Date created: February 21st, 2013
 * Last date modified: May 10th, 2013
 *
 * Sources Used:
 *      None
 */

#include <string>

using namespace std;

#ifndef TRADEITEM_H
#define TRADEITEM_H

class TradeItem
{
    public:

    /***** constructor/destructor declarations *****/

        /**
         * Default constructor for the TradeItem class.
         * @param strItemName The name of the item (default is the string EMPTY)
         * @param uintItemValue The value of the item (default is the integer 0)
         * @param boolIsContraband Flag to set whether the item is legal/illegal (default is boolean false)
         */
        TradeItem( string strItemName = "EMPTY", unsigned int uintItemValue = 0, bool boolIsContraband = false );

        /**
         * Destructor for the TradeItem class.  This destructor does not perform any operations.
         */
        ~TradeItem();

    /***** overloaded operator declarations *****/

        /**
         * Overloaded comparison operator.  Checks to see whether the values for the variables strItemName and
         * uintItemValue are the same between two instances of TradeItem.
         * @param objTradeItemLeft The left instance of TradeItem to be compared.
         * @param objTradeItemRight The right instance of TradeItem to be compared.
         */
        friend bool operator == ( const TradeItem& objTradeItemLeft, const TradeItem& objTradeItemRight );

    /***** public function declarations *****/
```

```
51
52          /**
53           * Accessor function for the strItemName variable.
54           * @returns string object with the current value of the strItemName variable.
55           */
56          string  getItemName();
57
58          /**
59           * Mutator function for the strItemName variable.
60           * @param strItemName The new string value to be assigned to the strItemName variable.
61           */
62          void    setItemName( string strItemName );
63
64          /**
65           * Accessor function for the uintItemValue variable.
66           * @returns unsigned int value for the strItemName variable.
67           */
68          unsigned int    getItemValue();
69
70          /**
71           * Mutator function for the uintItemValue variable.
72           * @param uintItemValue The new unsigned int value to be assigned to the uintItemValue variable.
73           */
74          void            setItemValue( unsigned int uintItemValue );
75
76          /**
77           * Accessor function for the boolIsContraband variable.
78           * @returns bool value for the boolIsContraband variable.
79           */
80          bool    getIsContraband();
81
82          /**
83           * Mutator function for the boolIsContraband variable.
84           * @param boolIsContraband The new boolean value to be assigned to the boolIsContraband variable.
85           */
86          void    setIsContraband( bool boolIsContraband );
87
88      private:
89
90      /***** private variable declarations *****/
91          string strItemName;
92          unsigned int uintItemValue;
93          bool boolIsContraband;
94
95  };
96
97  #endif
98
```