

```

1  /*
2   * Class name: SmugglerShip (header file)
3   * Class description: Class definition for the type SmugglerShip. SmugglerShip objects represent a ship for a game
4   *                     having a captain, cargo holds, money, ship name, etc. Instances of this class are used to define different
5   *                     types of ships with unique properties and attributes.
6   *
7   * Programmer: Chad Philip Johnson
8   * Date created: February 21st, 2013
9   * Last date modified: May 10th, 2013
10  *
11  * Sources Used:
12  *     tradeitem.h
13  *         - for accomodating created instances of the TradeItem class which the ship will store in its cargo holds
14  */
15
16  #include <string>
17
18  #include "tradeitem.h"
19
20  using namespace std;
21
22  #ifndef SMUGGLERSHIP_H
23  #define SMUGGLERSHIP_H
24
25  class SmugglerShip
26  {
27  public:
28
29      /***** constructor/destructor declarations *****/
30
31      /**
32       * Default constructor for the SmugglerShip class. Performs the following assignments: the string "No Name" to the
33       * variable strCaptainName,
34       * the string "SS Smuggler" to the variable strShipName, the value 5 to the variable uintLegalCargoCapacity, a dynamic
35       * array of size
36       * uintLegalCargoCapacity to the pointer objTradeItemLegalCargo, the value 5 to the variable uintLegalCargoCapacity, a
37       * dynamic array of size
38       * uintIllegalCargoCapacity to the pointer objTradeItemIllegalCargo, the value 0 to the variable uintLegalCargoUsed, the
39       * value 0 to the
40       * variable uintIllegalCargoUsed, the value 100 to the variable uintAstros.
41       */
42      SmugglerShip();
43
44      /**
45       * Overloaded constructor for the SmugglerShip class.
46       * @param strCaptainName string representing the name of the captain of the ship.
47       * @param strShipName string representing the name of the ship.
48       * @param uintLegalCargoCapacity unsigned int representing the maximum legal cargo capacity of the ship.
49       * @param uintIllegalCargoCapacity unsigned int representing the maximum illegal cargo capacity of the ship.
50       * @param uintAstros unsigned int representing the amount of astros (or currency) onboard the ship.

```

```

47     */
48     SmugglerShip( string strCaptainName, string strShipName, unsigned int uintLegalCargoCapacity, unsigned int
uintIllegalCargoCapacity, unsigned int uintAstros );
49
50     /**
51     * Destructor for the SmugglerShip class. Frees the memory associated with the objTradeItemLegalCargo and
objTradeItemIllegalCargo pointers.
52     */
53     ~SmugglerShip();
54
55     /***** public function definitions *****/
56
57     /**
58     * Adds astros (or currency) to the current number of astros held by the ship.
59     * @param uintNumberOfAstros unsigned int representing the amount of astros to be added to the current total.
60     */
61     void addAstros( unsigned int uintNumberOfAstros );
62
63     /**
64     * Remove an amount of astros from the current number of astros held by the ship.
65     * @param uintNumberOfAstros unsigned int representing the number of astros to be removed from the ship.
66     * @return true when the ship has more astros than the requested amount; false when the ship does not have enough.
67     */
68     bool spendAstros( unsigned int uintNumberOfAstros );
69
70     /**
71     * Checks the supplied cargo type and returns the ships maximum capacity for that cargo type. (NOTE: This function
allows for additional
72     * conditions to be checked before calling the private function fulfillCargoCapacity() which fulfills the request.
Currently no
73     * additional conditions have been implemented.)
74     * @param charCargoType Character value representing the type of cargo to be checked; available values are 'i' and 'I' for
75     * illegal cargo, 'l' and 'L' for legal cargo
76     * @return unsigned int value representing the maximum capacity for a cargo type.
77     */
78     unsigned int getCapacity( const char &charCargoType ) const;
79
80     /**
81     * Add object of type TradeItem to the ship. (NOTE: This function allows for additional conditions to be checked
before calling the private
82     * function fulfillAddCargo() which fulfills the request. Currently no additional conditions have been implemented.)
83     * @param objTradeItemCargoItem The instance of TradeItem to be added to the ship's cargo.
84     * @param charCargoType Character value representing the type of cargo that is being added; available values are 'i' and
'I' for
85     * illegal cargo, 'l' and 'L' for legal cargo
86     * @return true when addition of cargo was successful (room available); false on failure (no room available)
87     */
88     bool addCargo( const TradeItem &objTradeItemCargoItem, const char &charCargoType );
89
90     /**

```

```

91     * Checks to see whether a cargo space in the ship is filled or vacant. (NOTE: This function allows for additional
92     * conditions to be checked
93     * before calling the private function fulfillCheckCargo() which fulfills the request. Currently no additional
94     * conditions have been implemented.)
95     * @param uintCargoIndex unsigned int value representing the array index to be checked.
96     * @param charCargoType char representing the type of cargo to be checked; available values are 'i' and 'I' for illegal
97     * cargo, 'l' and 'L' for
98     * legal cargo
99     * @return the instance of TradeItem held at that location.
100    */
101    TradeItem& checkCargo( const unsigned int &uintCargoIndex, const char &charCargoType );
102
103    /**
104     * Removes cargo from the ship. (NOTE: This function allows for additional conditions to be checked before calling the
105     * private function
106     * fulfillRemoveCargo() which fulfills the request. Currently no additional conditions have been implemented.)
107     * @param uintCargoIndex unsigned int value representing the array index to be checked.
108     * @param charCargoType char representing the type of cargo to be checked; available values are 'i' and 'I' for illegal
109     * cargo, 'l' and 'L' for
110     * legal cargo
111     * @return the instance of TradeItem that was removed from the ship.
112     */
113    TradeItem removeCargo( const unsigned int &uintCargoIndex, const char &charCargoType );
114
115    /**
116     * Accessor function for the strCaptainName variable. Retrieve the name of the ship's captain.
117     * @return string value for the variable strCaptainName.
118     */
119    string getCaptainName() const;
120
121    /**
122     * Mutator function for the strCaptainName variable. Change the name of the ship's captain.
123     * @param strCaptainName The new string value for the name of the ship's captain.
124     */
125    void setCaptainName( string strCaptainName );
126
127    /**
128     * Accessor function for the strShipName variable. Retrieve the name of the ship.
129     * @return string value for the variable strShipName
130     */
131    string getShipName() const;
132
133    /**
134     * Mutator function for the strShipName variable. Change the name of the ship.
135     * @param strShipName The new string value for the name of the ship.
136     */
137    void setShipName( string strShipName );

```

```

136     /**
137     * Accessor function for the uintAstros variable. Retrieve the current number of astros held by the ship.
138     * @return unsigned int value for the variable uintAstros.
139     */
140     unsigned int getAstros() const;
141
142     /**
143     * Mutator function for the uintAstros variable. Change the current number of astros held by the ship.
144     * @param uintAstros The new unsigned int value for the amount of astros held by the ship.
145     */
146     void setAstros( unsigned int uintAstros );
147
148 private:
149
150     /***** private function declarations *****/
151
152     /**
153     * Reports the ships maximum storage capacity for either legal or illegal cargo.
154     * @param charCargoType The type of of cargo to check; acceptable values are 'i' and 'I' for illegal cargo, 'l' and 'L'
155     * for legal cargo
156     * @return unsigned int value for the variable uintIllegalCargoCapacity or uintLegalCargoCapacity.
157     */
158     unsigned int fulfillCargoCapacity( const char &charCargoType ) const;
159
160     /**
161     * Checks to see whether there is room in the ship for a new piece of cargo to be added. Adds the cargo only if room
162     * exists.
163     * @param objTradeItemCargoItem TradeItem object to be added to the ship.
164     * @param charCargoType char value to specify which type of cargo to add; acceptable values are 'i' and 'I' for illegal
165     * cargo, 'l' and 'L'
166     * for legal cargo
167     * @return true of addition of cargo was successful, false if not.
168     */
169     bool fulfillAddCargo( const TradeItem &objTradeItemCargoItem, const char &charCargoType );
170
171     /**
172     * Reports whether a cargo storage location is filled or vacant.
173     * @param uintCargoIndex Storage location to check.
174     * @param charCargoType The type of cargo to check; acceptable values are 'i' and 'I' for illegal cargo, 'l' and 'L' for
175     * legal cargo
176     * @return The TradeItem object if the storage location contains cargo, or a "junk" item if the space is empty or does
177     * not exist.
178     */
179     TradeItem& fulfillCheckCargo( const unsigned &uintCargoIndex, const char &charCargoType );
180
181     /**
182     * Remove cargo from a storage location within the ship if that location contains cargo.
183     * @param uintCargoIndex Storage location to check.
184     * @param charCargoType The type of cargo to check; acceptable values are 'i' and 'I' for illegal cargo, 'l' and 'L' for
185     * legal cargo

```

```
180         * @return The TradeItem object if the storage location contains cargo, or a "junk" item if the space is empty or does
181         not exist.
182         */
183         TradeItem fulfillRemoveCargo( const unsigned &uintCargoIndex, const char &charCargoType );
184
185     /***** private variable declarations *****/
186     string strCaptainName, strShipName;
187     unsigned int uintAstros, uintLegalCargoCapacity, uintLegalCargoUsed, uintIllegalCargoCapacity, uintIllegalCargoUsed;
188     TradeItem objTradeItemJunk, *objTradeItemLegalCargo, *objTradeItemIllegalCargo;
189 };
190 #endif
191
```