```cpp
/*
 * Class name:  SmugglerShip (source file)
 * Class description:  Class definition for the type SmugglerShip.  SmugglerShip objects represent a ship for a game
 *      having a captain, cargo holds, money, ship name, etc.  Instances of this class are used to define different
 *      types of ships with unique properties and attributes.
 *
 * Programmer:  Chad Philip Johnson
 * Date created: February 21st, 2013
 * Last date modified: May 10th, 2013
 *
 * Sources Used:
 *      tradeitem.h
 *          - for accomodating created instances of the TradeItem class which the ship will store in its cargo holds
 */

#include "smugglership.h"
/* constructor/destructor definitions */

SmugglerShip::SmugglerShip()
: strCaptainName( "No Name" ), strShipName( "SS Smuggler" ), uintLegalCargoCapacity( 5 ), uintIllegalCargoCapacity( 3 ), uintLegalCargoUsed( 0 ), uintIllegalCargoUsed( 0 ), uintAstros( 100 )
{
    objTradeItemLegalCargo      = new TradeItem[uintLegalCargoCapacity];
    objTradeItemIllegalCargo    = new TradeItem[uintIllegalCargoCapacity];
    objTradeItemJunk            = TradeItem( "Junk", 0, false);
}

SmugglerShip::SmugglerShip( string strCaptainName, string strShipName, unsigned int uintLegalCargoCapacity, unsigned int uintIllegalCargoCapacity, unsigned int uintAstros )
: strCaptainName( strCaptainName ), strShipName( strShipName ), uintLegalCargoCapacity( uintLegalCargoCapacity ), uintIllegalCargoCapacity( uintIllegalCargoCapacity ), uintAstros( uintAstros ), uintLegalCargoUsed( 0 ), uintIllegalCargoUsed( 0 )
{
    objTradeItemLegalCargo      = new TradeItem[uintLegalCargoCapacity];
    objTradeItemIllegalCargo    = new TradeItem[uintIllegalCargoCapacity];
    objTradeItemJunk            = TradeItem( "Junk", 0, false);
}

SmugglerShip::~SmugglerShip()
{
    delete [] this->objTradeItemLegalCargo;
    delete [] this->objTradeItemIllegalCargo;
}

/* public function definitions */

void SmugglerShip::addAstros( unsigned int uintNumberOfAstros )
{
    setAstros( (uintAstros + uintNumberOfAstros) );
}

```

```
48  bool SmugglerShip::spendAstros( unsigned int uintNumberOfAstros )
49  {
50      if( uintNumberOfAstros <= getAstros() )
51      {
52          setAstros( (uintAstros - uintNumberOfAstros) );
53          return true;
54      } else {
55          return false;
56      }
57  }
58
59  unsigned int SmugglerShip::getCapacity( const char &charCargoType ) const
60  {
61      return this->fulfillCargoCapacity( charCargoType );
62  }
63
64  bool SmugglerShip::addCargo( const TradeItem &objTradeItemCargoItem, const char &charCargoType )
65  {
66      return this->fulfillAddCargo( objTradeItemCargoItem, charCargoType );
67  }
68
69  TradeItem& SmugglerShip::checkCargo( const unsigned &uintCargoIndex, const char &charCargoType )
70  {
71      return this->fulfillCheckCargo( uintCargoIndex, charCargoType );
72  }
73
74  TradeItem SmugglerShip::removeCargo( const unsigned &uintCargoIndex, const char &charCargoType )
75  {
76      return this->fulfillRemoveCargo( uintCargoIndex, charCargoType );
77  }
78
79  /* private function definitions */
80
81  unsigned int SmugglerShip::fulfillCargoCapacity( const char &charCargoType ) const
82  {
83      switch( charCargoType )
84      {
85          case 'L':
86          case 'l':
87              return uintLegalCargoCapacity;
88              break;
89
90          case 'I':
91          case 'i':
92              return uintIllegalCargoCapacity;
93              break;
94
95      }
96
97      return 0;
```

```
98   }
99
100  bool SmugglerShip::fulfillAddCargo( const TradeItem &objTradeItemCargoItem, const char &charCargoType )
101  {
102
103      switch( charCargoType )
104      {
105          case 'L':
106          case 'l':
107              if( uintLegalCargoUsed < uintLegalCargoCapacity )
108              {
109                  objTradeItemLegalCargo[uintLegalCargoUsed] = objTradeItemCargoItem;
110                  uintLegalCargoUsed++;
111                  return true;
112              }
113              else
114              {
115                  return false;
116              }
117
118              break;
119
120          case 'I':
121          case 'i':
122              if( uintIllegalCargoUsed < uintIllegalCargoCapacity )
123              {
124                  objTradeItemIllegalCargo[uintIllegalCargoUsed] = objTradeItemCargoItem;
125                  uintIllegalCargoUsed++;
126                  return true;
127              }
128              else
129              {
130                  return false;
131              }
132              break;
133      }
134  }
135
136  TradeItem& SmugglerShip::fulfillCheckCargo( const unsigned &uintCargoIndex, const char &charCargoType )
137  {
138      switch( charCargoType )
139      {
140          case 'L':
141          case 'l':
142              if( uintCargoIndex < uintLegalCargoCapacity )
143              {
144                  return objTradeItemLegalCargo[uintCargoIndex];
145              }
146              break;
147
```

```
148         case 'I':
149         case 'i':
150             if( uintCargoIndex < uintIllegalCargoCapacity )
151             {
152                 return objTradeItemIllegalCargo[uintCargoIndex];
153             }
154             break;
155
156     }
157
158     return objTradeItemJunk;
159
160 }
161
162 TradeItem SmugglerShip::fulfillRemoveCargo( const unsigned &uintCargoIndex, const char &charCargoType )
163 {
164     TradeItem objTradeItemSelectedItem;
165
166     switch( charCargoType )
167     {
168         case 'L':
169         case 'l':
170             if( uintCargoIndex < uintLegalCargoCapacity)
171             {
172                 if( uintLegalCargoUsed > 0 )
173                     uintLegalCargoUsed--;
174                 objTradeItemSelectedItem = objTradeItemLegalCargo[uintCargoIndex];
175
176                 objTradeItemLegalCargo[uintCargoIndex]  = TradeItem();
177
178                 return objTradeItemSelectedItem;
179             }
180
181             break;
182
183         case 'I':
184         case 'i':
185             if( uintCargoIndex < uintIllegalCargoCapacity )
186             {
187                 uintIllegalCargoUsed--;
188                 objTradeItemSelectedItem = objTradeItemIllegalCargo[uintCargoIndex];
189
190                 objTradeItemIllegalCargo[uintCargoIndex]    = TradeItem();
191
192                 return objTradeItemSelectedItem;
193             }
194
195             break;
196
197     }
```

```
198
199         return objTradeItemJunk;
200
201     }
202
203
204     /* accessor/mutator function definitions */
205
206     string SmugglerShip::getCaptainName() const
207     {
208         return this->strCaptainName;
209     }
210
211     void SmugglerShip::setCaptainName( string strCaptainName )
212     {
213         this->strCaptainName = strCaptainName;
214     }
215
216     string SmugglerShip::getShipName() const
217     {
218         return this->strShipName;
219     }
220
221     void SmugglerShip::setShipName( string strShipName )
222     {
223         this->strShipName = strShipName;
224     }
225
226     unsigned int SmugglerShip::getAstros() const
227     {
228         return this->uintAstros;
229     }
230
231     void SmugglerShip::setAstros( unsigned int uintAstros )
232     {
233         this->uintAstros = uintAstros;
234     }
235
```