```java
/*
 * Programmer:  Chad Philip Johnson
 * Date Created:  Thursday, November 06th, 2012
 * Date of Last Modification:  Tuesday, December 11th, 2012
 *
 * Description:
 * Player.class represents the player's character details and items held.
 * Player.class depends on Item.class and its subclasses Weapon.class and
 * Armor.class for player  inventory.
 */

import java.util.*;
import java.io.Serializable;

/**
 * Player.class represents the player's character details and items held.
 * Player.class depends on Item.class and its subclasses Weapon.class and
 * Armor.class for player  inventory.
 *
 * @author Chad Philip Johnson
 * @version 1.0
 */

public class Player implements Serializable {

    String strPlayerName;
    int intGoldHeld;
    ArrayList<Item> itemsHeld;

    /**
      * Default constructor:
      * Set the player's name, amount of gold and items held to their default values.
      */

    public Player( String strPlayerName )   {

        this.strPlayerName  = strPlayerName;
        intGoldHeld         = 0;
        itemsHeld           = new ArrayList<Item>();

    }

    /**
     * Remove a currently held item from inventory.
     *
     * @param intIndex Numerical value representing the item to be removed.
     * @return Item Item that was removed.
     */

    public Item removeItem( int intIndex )  {   return itemsHeld.remove( intIndex );    }
```

```
51
52      /**
53       * Add an item to the player's inventory.
54       *
55       * @param itemToAdd The item object that will be added to the player's inventory.
56       */
57
58      public void addItem( Item itemToAdd )   {   itemsHeld.add( itemToAdd ); }
59
60      /**
61       * Adds gold to the player's purse.
62       *
63       * @param intGoldAmount The amount of gold pieces to add to the player's purse.
64       */
65
66      public void addGold( int intGoldAmount )    {   this.intGoldHeld    += intGoldAmount;   }
67
68      /**
69       * Prints details about the player such as name, number of gold pieces, and currently held items.
70       *
71       * @return String A message to the console containing this information.
72       */
73
74      public String toString()    {
75
76          String strPlayerDetails = "Character Name:  " + strPlayerName + "\n" + "Gold:  " + Integer.toString( intGoldHeld ) + "\n";
77
78          if( itemsHeld.isEmpty() != false )  {
79
80              strPlayerDetails += "You currently have the following items:\n";
81
82              for( Item item : itemsHeld )    {   strPlayerDetails += "\t" + item;    }
83
84          }
85
86          return strPlayerDetails;
87
88      }
89
90      // Accessor/Mutator methods
91
92      public String getPlayerName()   {   return this.strPlayerName;  }
93
94      public void setPlayerName( String strPlayerName )   {   this.strPlayerName  = strPlayerName;    }
95
96      public int getGoldHeld()    {   return this.intGoldHeld;    }
97
98      public void setGoldHeld( int intGoldHeld )  {   this.intGoldHeld    = intGoldHeld;  }
99
100     public ArrayList<Item> getItem()    {   return this.itemsHeld;  }
```

```
101
102
103
104   }
```