

## Woo

**private double balance**

keeps track of player balance

**private int gameType**

stores the current game type (numbers correspond to which game)

**private double bet**

store the player-determined bet for the game

**public void play()**

Prints instructions, selects game

**public void replay()**

method to play next game after you're done with one

**public void loseMoney()**

method to subtract money from balance

**public void winMoney()**

method to add money to balance

**public static void main(String[] args)**

runs play method

Cards (abstract)
<b>protected ArrayList&lt;Object&gt; deck = new ArrayList();</b>
<b>public void createDeck()</b> creates deck
HighLow
<b>Object dealerCard</b> Dealer's card
<b>Object playerCard</b> Player's card
Methods
<b>public static boolean play()</b> calls playHelp()
<b>public boolean playHelp()</b> runs the game
RussianRoulette
<b>public static boolean play()</b> 1/6 chance to die, 5/6 to live. Very simple code

blackjack
<b>private int dealerCount</b> Numerical value of dealer's hand
<b>private ArrayList dealerHand</b> ArrayList representation of dealer's hand
<b>private int playerCount</b> Numerical value of player's hand
<b>private ArrayList playerHand</b> ArrayList representation of player's hand
<b>ArrayList&lt;object&gt; deck</b> Represents a deck
Methods
<b>public static String play()</b> called by Woo, invokes playHelp()
<b>public String playHelp()</b> sets up the game, prints info to terminal, interprets keyboard input and runs the game
<b>public Object pickcard()</b> picks a card from the deck
<b>public boolean blackjack(ArrayList)</b> Sees if the given array is a blackjack
<b>public void createDeck()</b> Created the deck
<b>public int countCards(ArrayList)</b> Adds up the cards in the given arraylist
<b>public Object hit()</b> Gives a card to object calling it

Hangman
<b>private static final String FILENAME = "Words.txt"</b> file containing words for hangman
<b>private ArrayList&lt;String&gt; Words</b> String of words read in from Words.txt
<b>private ArrayList&lt;String&gt; playerLetters</b> Letter's that the player has guessed right
<b>private ArrayList&lt;String&gt; playerGuesses</b> The letters the player has guessed so far
<b>private String word</b> The hangman Word
<b>private int tries</b> The number of incorrect guesses the player has made
Methods
<b>public static boolean play()</b> calls playHelp()
<b>public boolean playHelp()</b> Runs the game, returns boolean
<b>public boolean win()</b> Checks if PlayerLetters is the same as the word, in which case they win
<b>public void readWords()</b> Method to read words from .txt file into an ArrayList

ConnectFour
<b>private int[][] Playboard</b> Board that is played on
<b>private boolean Victory</b> Turns true when player wins
<b>private int[] Available</b> Sees what columns are available for placement
<b>private int winner</b> Holds value of winner
Methods
<b>public void connectFour()</b> creates a board with 0 in each slot
<b>public String toString()</b> prints out the board for the user to see
<b>private int[] place(int column, int player)</b> places the pieces
<b>public String toString(int[][] board)</b> Prints board
<b>public static String toString(int[] intArray)</b>
<b>public void computerMove()</b> determines next computer move using AI
<b>public static boolean play()</b> Method called by Woo, plays the game
<b>public void wipe()</b> Resets everything
<b>public boolean isGameOver()</b> checks to see if there is a four in a row
<b>public int findThree()</b> AI for the computer so that it find three in row to place its next move, returns the value of the index of the next move
<b>public int findTwo()</b> AI for the computer to find two in a row so it can place its moves strategically, returns index of next move