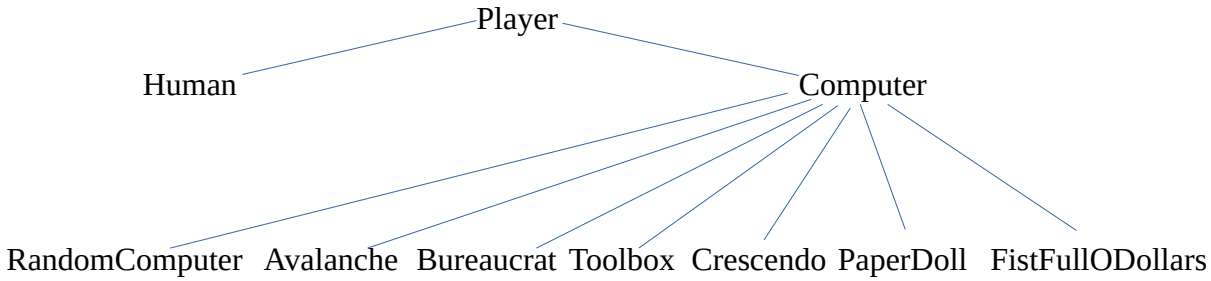
Assignment 3 Design

By 1686498

Hierarchy:



Player
string name – stores name
String getName – returns name void setName – sets the name

- Top of our hierarchy
- Has a name so can be identified

Human
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void setMoves – sets moves according to input void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Has set moves function since it will take user input

Computer
Maybe a virtual function???

- H

Assignment 3 Design

By 1686498

Avalanche
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Always chooses rock

Bureaucrat
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Always chooses paper

Crescendo
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Chooses move in the order paper, scissors, rock

Assignment 3 Design

By 1686498

FistFullODollars
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Chooses moves in the order rock, paper, paper

PaperDoll
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Chooses moves in the order paper, scissors, scissors

RandomComputer
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins void setMove – chooses a random move

- Chooses a random move

Assignment 3 Design

By 1686498

Toolbox
String moves – string containing moves int numberOfWins – stores the number of wins
Char getMoves – returns move void incrementNumberOfWins – adds 1 to the win count or resets the wins count int getNumberOfWins – returns the number of wins

- Always chooses scissors

All above classes:

- have a moves state so that the moves of a certain class can be stored
- have a numberOfWins state which allows for easy comparison to see who wins
- have a behaviour which allows me to access the move they wish to play
- have a way to increment the number of wins during the round and then reset the number of wins for the next round
- have a way to access numberOfWins to allow for comparison

Referee
Int rounds – will be 5
Player round – will take in 2 players and return the winner of that round

- Sets the number of rounds to be 5
- plays the round and determines who is the winner of that round

Tournament
Player winner – stores the winner vector round 1 – stores all players vector round 2 – stores winning players vector round 3 – stores last 2 players
Player driver – will take in all 8 players and will return winner player getWinner – returns the winner

- Has the winner stored in memory
- stores winners of each round as we go through
- driver will run and take in all the players and the referee and will then return who the winner is, will run referee behaviour to see who wins

Assignment 3 Design

By 1686498

Main:

- will create a vector of player objects
- will create a string for the input
- will check the string and create each different player based on input and then put them in the vector
- the player vector will then be inputted into tournament
- the driver will then be run
- the winner will be outputted

Testing:

I will test for a few different inputs

Input 1:

Avalanche Bureaucrat Bureaucrat Toolbox Toolbox Crescendo Crescendo FistFullODollars

Output 1:

Toolbox