Assignment 2 Design

By 1686498

human
String moves – stores players moves
Void setMoves – sets players moves char getMoves – returns players moves

- The users input will get taken into setMoves and only the moves will be stored in moves
- setMoves will remove the first 2 characters which will be a number and a space and then copy from there until the end
- It will check first though to make sure that the string contains something
- getMoves will return the string of moves

computer
char move – stores computers move
char getMove – returns computers move

- Move will store rock for this program cause the computer is dumb
- getMove will return the move rock

referee
Int numberOfGames – stores the number of games
Void playGame – Takes in a player and computer and decides who wins by comparing their moves void setNumberOfGames – take the users input and set the number of games int getNumberOfGames – will return the number of games

- playGame will take in who is playing and decide who wins
- It then prints out win,lose or tie depending on how the player went
- numberOfGames will will store the amount of games wanting to be played by the player
- setNumberOfGames will convert the users input to an int and store it in numberOfGames
- getNumberOfGames will return the number of games wanting to be played

GameContoller
Void play – Takes in the all the players playing

- play will take in all players playing and will call the referee function playGame inside of it

Main:

Assignment 2 Design

By 1686498

- Will take input from users
- will initialise all classes
- will set the number of games and players moves
- it will then call the play function in GameController

Testing:

I will test for some different samples of inputs

Input 1:

    0

Output 1:

    Enter in the form of k, followed by k moves with a space inbetween.

Input 1:

    1 R

Output 1:

    T

Input 1:

    1 P

Output 1:

    W

Input 1:

    1 S

Output 1:

    L

Input 1:

    3 S P R

Output 1:

    L W T

Input 1:

    4 R R P S

Output 1:

    T T W L

Input 1:

    2RS

Output 1:

    Enter in the form of k, followed by k moves with a space inbetween.

Input 1:

    -1

Output 1:

    Enter in the form of k, followed by k moves with a space inbetween.

Input 1:

Output 1:

    Enter in the form of k, followed by k moves with a space inbetween.