#Part 1

#Declares that the number trying to be guessed is 10 and there have been no attempts

number = 10

numAttempts = 0

#Asks the user what their name is and outputs a friend hello

username = input("What is your name? ")

print ("Hello", username,end=".") #end= places the string in parenthesis at the end of the printed statement

#Creates the numberguess function

def numberguess():

#User is asked to guess a number and the number is converted to an integer

userguess1 = input(" Guess a number between 1 and 20. ")

guess = int(userguess1)

#global allows you to access the outer scope variable

global numAttempts

#Adds creates a local variable for the number of attempts and adds 1, creates another variable called n from that result

numAttempts = numAttempts+1

n = numAttempts

#Prints a statement if the guess is correct

if guess == number:

print("Your guess is correct",username,"and you guessed it in", n, "attempt(s).")

#Prints a statement if the guess is too low

elif guess < number:

print("Your guess is too low.")

#Checks how many guesses there have beeen

if n < 3:

numberguess()

else:

print("Your three guesses are over, the number I was thinking of was 10.")

#Prints a statement if the guess is too high

elif guess > number:

print("Your guess is too high.")

#Checks how many guesses there have been

if n < 3:

numberguess()

else:

print("Your three guesses are over, the number I was thinking of was 10")

#Calls the numberguess function and adds 1 to the number of attempts

numberguess()

#

#

#

#

#

#

#

#

#

#

#

#

#

#Part 2

#Used this page to help me understand the logic and get my code started

#http://stackoverflow.com/questions/21540123/basic-rock-paper-scissors-in-python-only-player-2-is-winning

#Creates the player score variables and starts them at 0

player1score = 0

player2score = 0

#Creates the rockpaperscissors function

def rockpaperscissors():

#Asks each player for their input and stores it locally as a string

player1input = input("Player 1: Please enter either (R)ock, (P)aper, or (S)cissors: ")

player2input = input("Player 2: Please enter either (R)ock, (P)aper, or (S)cissors: ")

#Checks if the strings entered match up, if so, prints an string and returns a value of 0

if ((player1input in "rR") and (player2input in "rR")) or ((player1input in "sS") and (player2input in "sS")) or ((player1input in "pP") and (player2input in "pP")):

print("It's a tie.")

return 0

#Checks if player 1 wins, if so, prints a string and returns a value of 1

elif ((player1input in "rR") and (player2input in "sS")) or ((player1input in "pP") and (player2input in "rR")) or ((player1input in "sS") and (player2input in "pP")):

print("Player 1 wins.")

return 1

#Checks if player 2 wins, if so, prints a string and returns a value of 2

elif ((player1input in "rR") and (player2input in "pP")) or ((player1input in "pP") and (player2input in "sS")) or ((player1input in "sS") and (player2input in "rR")):

print("Player 2 wins.")

return 2

#If a string doesn't match (r R p P s S), this prints a string asks them to enter a proper letter

else:

print("Please use r or R for rock, p or P for paper, and s or S for scissors.")

#For loop is called 5 times

for i in range(5):

#Creates a variable named winner that is equal to the return from the function rockpaperscissors, this starts the rockpaperscissors function

winner = rockpaperscissors()

#If the rockpaperscissors function returns 0, this prints the score after it says "It's a tie"

if winner == 0:

print("Player 1 has", player1score,"point(s), and Player 2 has", player2score, "point(s).")

#If the rockpaperscissors function returns 1, this adds 1 to player1score and prints the score

elif winner == 1:

player1score = player1score + 1

print("Player 1 has", player1score,"point(s), and Player 2 has", player2score, "point(s).")

#If the rockpaperscissors function returns 2, this adds 1 to player2score and prints the score

elif winner == 2:

player2score = player2score + 1

print("Player 1 has", player1score,"point(s), and Player 2 has", player2score, "point(s).")