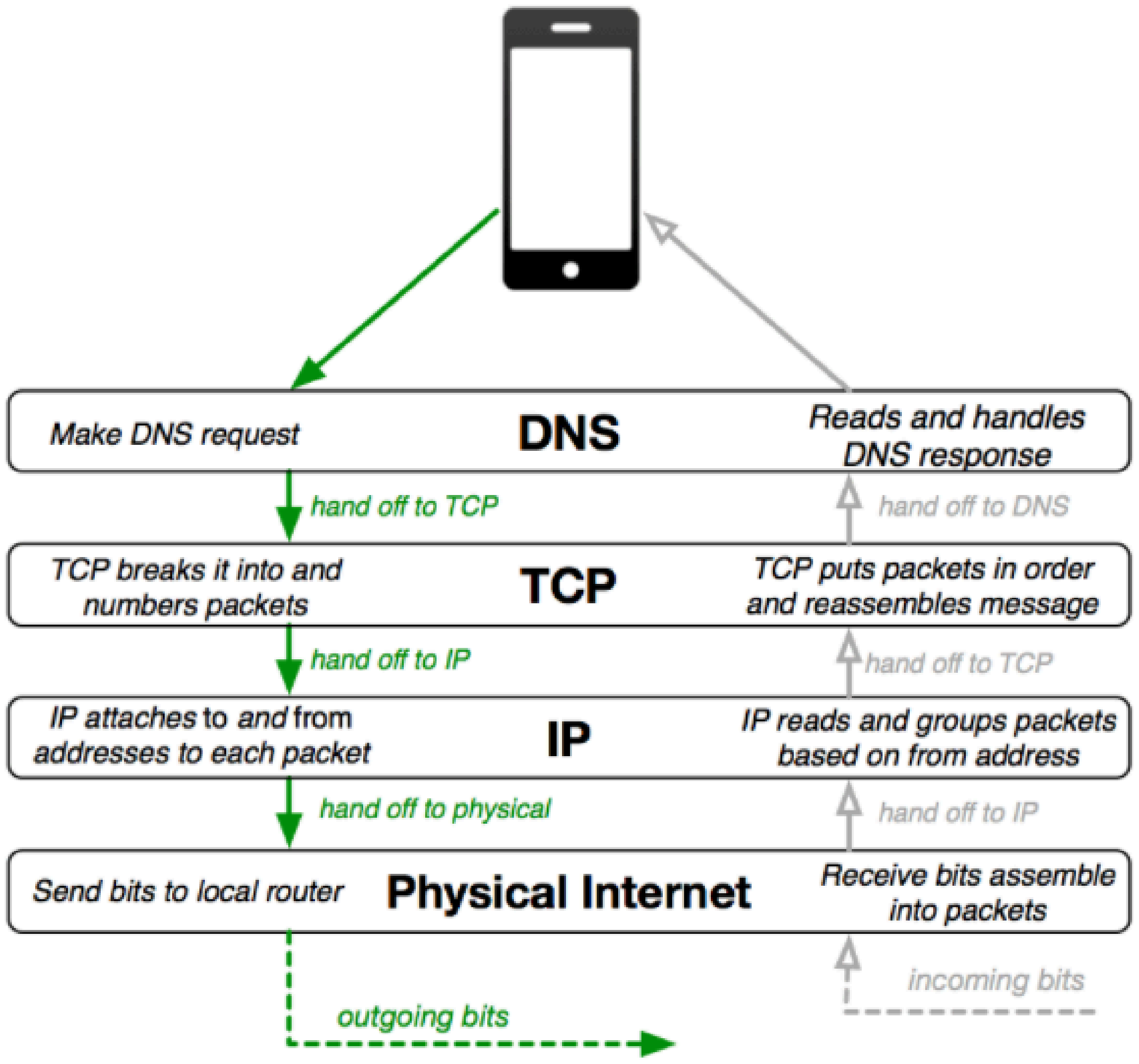
4/24/25

Packet- The binary information for a image,text or video and routing information IP address where it’s coming from and where it’s going and package limit
Routers- Mange where packets travel in the internet
Internet protocol- Choices best route for data to travel
Fault Tolerant- The internet can’t withstand some failures and still function
TCP-Transmission Control Protocol handles all of the sending a receiving of your data as packets
TCP and routers are scalable
HTTP-Hypertext Language Protocol
HTTP is the language used by computers to send information over the internet
HTML-HyperText
URL- Uniform Resource Locator
GET Request- gives HTML code
Post request-
Cookie- data the identifies
Secure Sockets Layer-SSL
Transport Layer Security-TLS
A layer of security around your communications
Certificate Authorities- verify identity of user

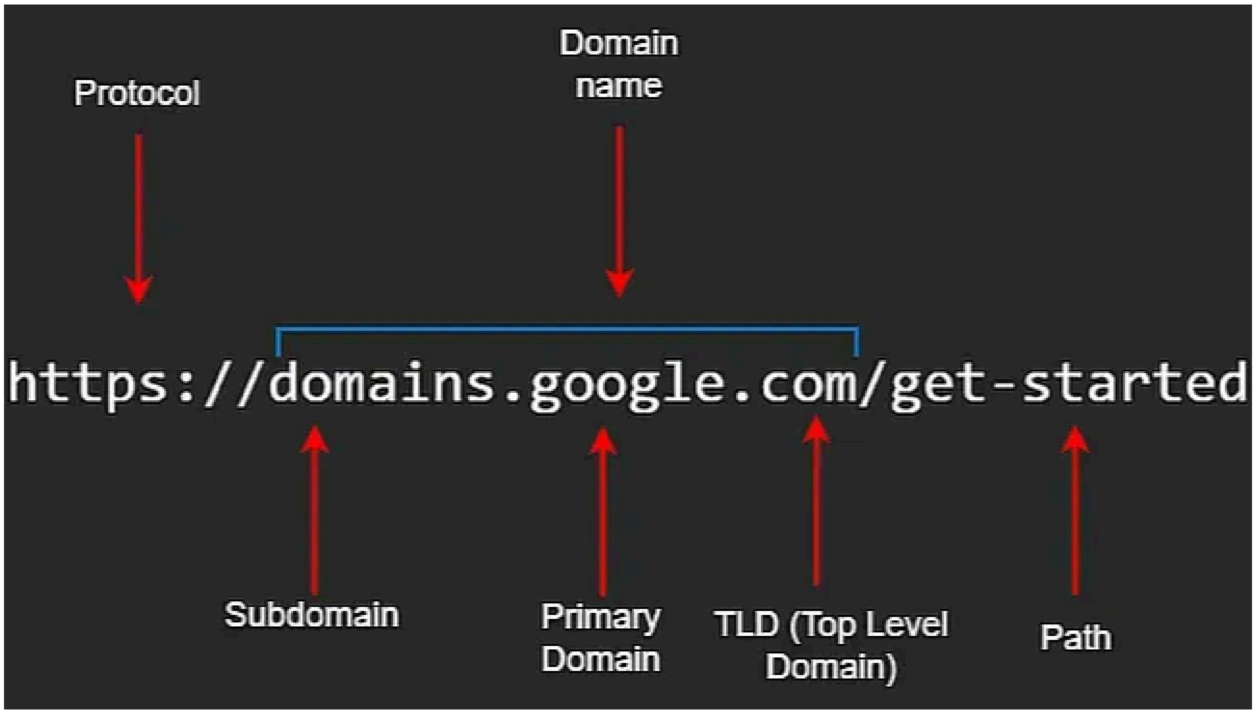

Make DNS request
DNS
Reads and handles DNS response
hand off to TCP
hand off to DNS
TCP breaks it into and numbers packets
TCP
TCP puts packets in order and reassembles message
hand off to IP
hand off to TCP
IP attaches to and from addresses to each packet
IP
IP reads and groups packets based on from address
hand off to physical
hand off to IP
Send bits to local router
Physical Internet
Receive bits assemble into packets
incoming bits
outgoing bits

4/22/25

IPV4:10.21.237.163
IPV6:fe80::ec65:a5f1:e523:5826%4

Internet is a network of networks that are interconnected
IP- Internet protocol
Protocol- Well-Known set of rules and standards used to communicate between machines
Each device a has a unique IP address to receive a data
Whenever a device signs into the internet it will be assigned a IP address
Hierarchy is a ranking system of
IP addresses are 32 bits long
The first numbers identify the country network the second number is the region network the third number is subnetwork and the 4th number is
IPv4 and IPv6 are the same

IPv6 have are 132 bits long
DNS- Domain Name System takes urls and looks up IP address to connect to devices
DNS servers are split in a distributed hierarchy of network
DNS was created to be an open and public communication protocol
DNS spoofing is when a hacker changes the a domain to a fake domain



4/17/25

FRQ # 2
Consider the first conditional statement included in the Procedure section of your Personalized Project Reference.

a. Describe your conditional statement, including its Boolean expression.

In my procedure, there is a conditional statement that takes the user's input of genre from a list of best selling video games and uses a loop to search the length of the list and selects a game that matches the users input example if genre==”action”etc Then it prints a random game that matches the correct genre.

b. Describe what the procedure does in general when the Boolean expression of this conditional statement evaluates to false.

When the boolean expression if genre ==”action” is false based on the users imputed genre, if it is false it takes the conditional if statement that is true based on the users input if genre==”puzzle”

4/4/25- CREATE Project

Python
```python
#DanielBrookins
#4/2/25
#CREATETask
#Lists of video game genre data sets
#This program use data sets from
https://en.wikipedia.org/wiki/List_of_best-selling_video_games
action
=["GrandTheftAutoV(Action/Adventure)","RedDeadRedemption2(Action/Adv
enture) ","TheWitcher3(Action/Adventure/RPG)
","TheElderScrollsV:Skyrim(Action/Adventure/RPG)
","SuperMarioBros.(Platformer) ","NewSuperMarioBros.(Platformer)
","NewSuperMarioBros.Wii(Platformer)
","The_Legend_of_Zelda:BreathoftheWild(Action/Adventure)
","SuperSmash ","Bros.Ultimate(Fighting)
","SuperMarioWorld(Platformer)
","SuperMarioBros.3(Platformer)","SonictheHedgehog(Action)
","DuckHunt(Shooter) ","BrainAge(Puzzle/Brain Training)"]
shooter =["PlayerUnknown'sBattlegrounds(Battle ","Royale/Shooter)
","CallofDuty:ModernWarfare3(First-Person ","Shooter)
","CallofDuty:Black ","Ops(First-Person ","Shooter)
","CallofDuty:BlackOpsII(First-Person ","Shooter)
","CallofDuty:ModernWarfare2(First-Person ","Shooter)
","CallofDuty:Ghosts(First-Person ","Shooter)"]
puzzle = ["Tetris(EAMobile)(Puzzle) ","Tetris(Nintendo)(Puzzle)
","Pac-Man(Arcade/Puzzle) ","Frogger(Arcade) ","Lemmings(Puzzle)"]
```

```
sports=["WiiSports(Sports) WiiFitandWiiFitPlus(Fitness/Exercise)
WiiSportsResort(Sports) FIFA18(Sports) MarioKartWii(Racing) 2
MarioKart8/Deluxe(Racing) MarioKartDS(Racing) MarioKart7(Racing)
KinectAdventures!(Party) WiiPlay(Party)"]
role playing=["PokémonRed/Green/Blue/Yellow(RPG)
","PokémonGold/Silver/Crystal(RPG)
","PokémonSun/Moon/UltraSun/UltraMoon(RPG)
","PokémonDiamond/Pearl/Platinum(RPG)
","PokémonRuby/Sapphire/Emerald(RPG) ","DiabloIII(ActionRPG)
","Borderlands2(ActionRPG)"]
platformer =["SuperMarioBros.(Platformer)
","SuperMarioLand(Platformer) ","SuperMarioWorld(Platformer)
","SuperMarioBros.3 ","(Platformer) ","NewSuperMario
","Bros.(Platformer) ","New ","SuperMario ","Bros.Wii(Platformer)"]

#imports random video game based on the users input
import random

# Searches through video games list to find a random game to
recommend to the user based on what genre they are interested in
filtered_list = []
# Greets the user by welcoming them to the program
print("Welcome to Best selling video game gere picker!")

# User input for best video game genres they are interested in
genre = input("What Best selling video game genres are you
interested in?(Action/Adventure,FPS,Puzzle,Sports,Role
Playing,Platformer)")
print("hmmm let me see what I can find")

def video_games(g):
        if genre == "action":
            print(random.choice(action))
        if genre == "shooter":
            print(random.choice(shooter))
        if genre == "puzzle":
            print(random.choice(puzzle))
        if genre == "sports":
            print(random.choice(sports))
        if genre == "roleplaying":
            print(random.choice(role playing))
        if genre == "platformer":
            print(random.choice(platformer))

video_games(genre)
```

Action/Adventure

- **Grand Theft Auto V** (Action/Adventure)
- **Red Dead Redemption 2** (Action/Adventure)
- **The Witcher 3** (Action/Adventure/RPG)
- **The Elder Scrolls V: Skyrim** (Action/Adventure/RPG)
- **Super Mario Bros.** (Platformer)
- **New Super Mario Bros.** (Platformer)
- **New Super Mario Bros. Wii** (Platformer)
- **The Legend of Zelda: Breath of the Wild** (Action/Adventure)

- **Super Smash Bros. Ultimate** (Fighting)
- **Super Mario World** (Platformer)
- **Super Mario Bros. 3** (Platformer)

Shooter

- **PlayerUnknown's Battlegrounds** (Battle Royale/Shooter)
- **Call of Duty: Modern Warfare 3** (First-Person Shooter)
- **Call of Duty: Black Ops** (First-Person Shooter)
- **Call of Duty: Black Ops II** (First-Person Shooter)
- **Call of Duty: Modern Warfare 2** (First-Person Shooter)
- **Call of Duty: Ghosts** (First-Person Shooter)

Puzzle

- **Tetris (EA Mobile)** (Puzzle)
- **Tetris (Nintendo)** (Puzzle)
- **Pac-Man** (Arcade/Puzzle)
- **Frogger** (Arcade)
- **Lemmings** (Puzzle)

Sports

- **Wii Sports** (Sports)
- **Wii Fit and Wii Fit Plus** (Fitness/Exercise)
- **Wii Sports Resort** (Sports)

- **FIFA 18** (Sports)
- **Mario Kart Wii** (Racing)
- **Mario Kart 8 / Deluxe** (Racing)
- **Mario Kart DS** (Racing)
- **Mario Kart 7** (Racing)
- **Kinect Adventures!** (Party)
- **Wii Play** (Party)

Role-Playing (RPG)

- **Pokémon Red / Green / Blue / Yellow** (RPG)
- **Pokémon Gold / Silver / Crystal** (RPG)
- **Pokémon Sun / Moon / Ultra Sun / Ultra Moon** (RPG)
- **Pokémon Diamond / Pearl / Platinum** (RPG)
- **Pokémon Ruby / Sapphire / Emerald** (RPG)
- **Diablo III** (Action RPG)
- **Borderlands 2** (Action RPG)

Platformer

- **Super Mario Bros.** (Platformer)
- **Super Mario Land** (Platformer)
- **Super Mario World** (Platformer)
- **Super Mario Bros. 3** (Platformer)
- **New Super Mario Bros.** (Platformer)

- **New Super Mario Bros. Wii** (Platformer)

Fighting

- **Super Smash Bros. Ultimate** (Fighting)

Simulation

- **Minecraft** (Sandbox/Survival)
- **Terraria** (Sandbox/Survival)
- **Nintendogs** (Simulation)
- **Wii Fit and Wii Fit Plus** (Fitness/Exercise)

Racing

- **Gran Turismo 4** (Racing)

Action

- **Sonic the Hedgehog** (Action)
- **Duck Hunt** (Shooter)
- **Brain Age** (Puzzle/Brain Training)

Open World

- **Grand Theft Auto IV** (Open-World/Action)
- **Grand Theft Auto: San Andreas** (Open-World/Action)
- **Grand Theft Auto: Vice City** (Open-World/Action)
- **Red Dead Redemption 2** (Open-World/Action)

4/2/25- CREATE Project FRQ

#1 Identify the expected group of users of your program. Explain how your program addresses at least one concern or interest of the users you identified.

Python
```python
#DanielBrookins
#4/2/25
#CREATETask
videogames=
[Minecraft","GrandTheftAutoV","Tetris(EAMobile)","WiiSports","Player
Unknown'sBattlegrounds","SuperMarioBros.","PokÃ©monRed/Green/Blue/Ye
llow","WiiFitandWiiFitPlus","Tetris(Nintendo)","Pac-Man","MarioKartW
ii","MarioKart8/Deluxe","WiiSportsResort","RedDeadRedemption2","NewS
uperMarioBros.","Terraria","NewSuperMarioBros.Wii","TheElderScrollsV
:Skyrim","DiabloIII","PokÃ©monGold/Silver/Crystal","DuckHunt","WiiPl
ay","TheWitcher3","GrandTheftAuto:SanAndreas","CallofDuty:ModernWarf
are3","CallofDuty:BlackOps","GrandTheftAutoIV","PokÃ©monSun/Moon/Ult
raSun/UltraMoon","PokÃ©monDiamond/Pearl/Platinum","CallofDuty:BlackO
psII","KinectAdventures!","FIFA18","SonictheHedgehog","Nintendogs","
MarioKartDS","CallofDuty:ModernWarfare2","PokÃ©monRuby/Sapphire/Emer
ald","Borderlands2","SuperMarioWorld","Frogger","Lemmings","GrandThe
ftAuto:ViceCity","TheLastofUs","TheLegendofZelda:BreathoftheWild","B
rainAge","SuperMarioBros.3","CallofDuty:Ghosts","SuperSmashBros.Ulti
mate","MarioKart7","SuperMarioLand","GranTurismo4"]
sales =
["2.00E+08","1.30E+08","1.00E+08","82900000","60000000","48240000","
47520000","43800000","43000000","39098000","37320000","33220000","33
130000","31000000","30800000","30300000","30300000","30000000","3000
```

```
0000","29490000","28300000","28020000","28000000","27500000","265000
00","26200000","25000000","24950000","24730000","24200000","24000000
","24000000","23982960","23960000","23600000","22700000","22540000",
"22000000","20972500","20000000","20000000","20000000","20000000","1
9080000","19010000","19000000","19000000","18840000","18710000","183
70500","17830000"]
Platforms =
["Multi-platform","Multi-platform","Mobile","Wii","Multi-platform","
Multi-platform","Multi-platform","Wii","GameBoy/NES","Multi-platform
","Wii","WiiU/Switch","Wii","Multi-platform","NintendoDS","Multi-pla
tform","Wii","Multi-platform","Multi-platform","GameBoyColor","NES",
"Wii","Multi-platform","Multi-platform","Multi-platform","Multi-plat
form","Multi-platform","Nintendo3DS","NintendoDS","Multi-platform","
Xbox360","Multi-platform","Multi-platform","NintendoDS","NintendoDS"
,"Multi-platform","GameBoyAdvance","Multi-platform","Multi-platform"
,"Multi-platform","Multi-platform","Multi-platform","PS3/PS4","Switc
h/WiiU","NintendoDS","Multi-platform","Multi-platform","NintendoSwit
ch","Nintendo3DS","Multi-platform","PS2/PSP"]
Initial Release =
["18-Nov-11","17-Sep-13","12-Sep-06","19-Nov-06","20-Dec-17","13-Sep
-85","27-Feb-96","1-Dec-07","14-Jun-89","Jul-80","10-Apr-08","29-May
-14","25-Jun-09","26-Oct-18","15-May-06","16-May-11","11-Nov-09","11
-Nov-11","16-May-12","21-Nov-99","21-Apr-84","2-Dec-06","19-May-15",
"26-Oct-04","8-Nov-11","9-Nov-10","29-Apr-08","18-Nov-16","28-Sep-06
","12-Nov-12","4-Nov-10","29-Sep-17","23-Jun-91","21-Apr-05","14-Nov
-05","10-Nov-09","21-Nov-02","18-Sep-12","21-Nov-90","5-Jun-81","14-
Feb-91","27-Oct-02","14-Jun-13","3-Mar-17","19-May-05","23-Oct-88","
5-Nov-13","7-Dec-18","1-Dec-11","21-Apr-89","28-Dec-04"]
Devloper =
["Mojang","Studios","Rockstar","North","EA","Mobile","Nintendo","EAD
","PUBG","Corporation","Nintendo","Game","Freak","Nintendo","EAD","N
intendo","R&D1","Namco","Nintendo","EAD","Nintendo","EAD","Nintendo"
```

```
,"EAD","Rockstar","Studios","Nintendo","EAD","Re-Logic","Nintendo","
EAD","Bethesda","Game","Studios","Blizzard","Entertainment","Game","
Freak","Nintendo","R&D1","Nintendo","EAD","CD","Projekt","Red","Rock
star","North","Infinity","Ward","/","Sledgehammer","Treyarch","Rocks
tar","North","Game","Freak","Game","Freak","Treyarch","Good","Scienc
e","Studio","EA","Canada","Sonic","Team","Nintendo","EAD","Nintendo"
,"EAD","Infinity","Ward","Game","Freak","Gearbox","Software","Ninten
do","EAD","Konami","DMA","Design","Rockstar","North","Naughty","Dog"
,"Nintendo","EPD","Nintendo","SPD","Nintendo","Infinity","Ward","Ban
dai","Namco","Studios","/","Sora","Ltd.","Nintendo","EAD","Nintendo"
,"R&D1","Polyphony","Digital"]
Publisher =
["Mojang","Studios","Rockstar","Games","Electronic","Arts","Nintendo
","PUBG","Corporation","Nintendo","Nintendo","Nintendo","Nintendo","
Namco","Nintendo","Nintendo","Nintendo","Rockstar","Games","Nintendo
","Re-Logic","/","505","Games","Nintendo","Bethesda","Softworks","Bl
izzard","Entertainment","Nintendo","Nintendo","Nintendo","CD","Proje
kt","Rockstar","Games","Activision","Activision","Rockstar","Games",
"Nintendo","/","The","PokÃ©mon","Company","Nintendo","/","The","PokÃ
©mon","Company","Activision","Xbox","Game","Studios","Electronic","A
rts","Sega","Nintendo","Nintendo","Activision","Nintendo","/","The",
"PokÃ©mon","Company","2K","Games","Nintendo","Sega","Psygnosis","Roc
kstar","Games","Sony","Computer","Entertainment","Nintendo","Nintend
o","Nintendo","Activision","Nintendo","Nintendo","Nintendo","Sony","
Computer","Entertainment"]
```

2/27/25- CREATE Requirements

1. Defined Function(s)w/Parameter
2. Input/Output
3. Iteration: Loop
4. Selection: If Statements
5. Array with data that is important to the program

2/20/25-Copyright

Copyright- a type of intellectual property that gives its owner the exclusive right to distribute,adapt,display and perform a creative work, usually for a limited time.

Creative Commons-

2/11/25- Return

A return statement marks the end of a function and specifies the value or values to pass back from the function.

-Can return any data type, including integers, floats, strings, list, dictionaries, and even other functions

Python
```python
#Daniel Brookins
#Code Review
#2/11/25
#Init
#Functions
def double_num(num):
    return(num*2)

# This function prints True if num is even, False otherwise

num=0
def is_even(num):
    if num % 2==0:
        return(True)
    else:
        return(False)

#Returns the last item in the list
numList= [1,2,3,4,5,6,7]
def last_item(list):

    return(list[len(list)-1])#Access one item from the list
#Main
last_item(numList)
double_num(10)
is_even(10)
```

Python

```python
num=0
def is_even(num):
    if num % 2==0:
        return(True)
    else:
        return(False)
#Main
print(is_even(10))
```

2/7/25 Images, Pixels and RGB

Pixel is a picture element
Each pixel is made up of Red Green and Blue lights
Resolution - the amount of pixels on your screen
RGB ranges from 0 to 255
0 is dark
255 is bright
Metadata - data that describes data
Width: 1 byte

Height: 1 byte

Bits per Pixel: 1 byte

n bits of pixel data

n = Width * Height * Bits per Pixel

2/3/25- Arrays with loops

Python
```python
#Array w/Loops
numberList = [1,61,43,23,13,5,68,98]
fruitList = ["cherry","banana","pear","apple","grape","watermelon"]

#Challenge 1
#Print the sum of all numbers in number list (variable)
sum=0
for number in numberList:
    sum += number
print(sum)

#Challenge 2
#print the largest number in numbers list
largestNum = 0
for number in numberList:
    if number > largestNum:
        largestNum = number
print(largestNum)
```

1/24/25

```
Python
# Main
try:
      x= int(input("Please enter a number 1-10:")
except ValueError:
      print("ERROR: Please enter a number!!")
```

1/22/25- Arrays

Index- position an item is in an array
index 0 is the first position

```
Python
#Daniel Brookins
#1/22/25
#List Methods
#Initialize
#mySchedule stores a list of the classes you are currently taking at
Jones  as strings
```

```python
#Initialize the list with your first three periods ONLY
mySchedule = ["College Algebra"," Honors Environmental Science","AP
CSP"]

print(mySchedule)

#Main

#Complete the following tasks using list/array methods. Print the
result for each task.

#Task 1: Append periods 4 - 7 to the list
mySchedule.append("Civil Engineering and Architecture")
mySchedule.append("Mandarin IV")
mySchedule.append("Digital Imaging II")
mySchedule.append("College Rhetoric and Composition")
mySchedule.append("ACLab")
print(mySchedule)

#Task 2: Insert your two lunch periods(A day and B Day) in their
appropriate location
mySchedule.insert(3,"C Lunch")
mySchedule.insert(8,"B Lunch")
print(mySchedule)
#Task 3: Remove your least favorite class
mySchedule.pop(1)
print(mySchedule)
#Task 4: Print just your 2nd period class by accessing the
appropriate index in your array
print(mySchedule[1])
```

```
#Task 5: Print only your A day schedule and then only your B day
schedule
print(mySchedule[0:4])
print(mySchedule[4:9])
```

1/6/25- Rock Paper Scissors

Python
```python
# = gets the value of
# == is equal to
```

12/6/24- Large Language Models

- Trained on the largest amount of data possible
- Uses random probabilities to predict based on what it has been trained on
- It needs Context
- Trained on neural networks
- Uses tokens word parts to predict sentence
- Needs human programing

12/4/24- Intro into Machine Learning (A.I)

- Machine learning
    - How computers recognize patterns and make decisions without being explicitly programmed

- Machine Learning learns from repetition of Data

.

12/2/24 - Strings

Python

```python
#Daniel Brookins
#12/2/24
#Strings

#Initialize
message = "computer science at Jones is the best?"

#Functions
#Main

#Complete the following tasks using string methods
#Task 1: Capitalize the first letter
x = message.capitalize()
print(x)
#Task 2: Uppercase the sentence( Use all capital letters)
x = message.upper()
print(x)
#Task 3: Replace the ? with an !
x = message.replace("?","!")
print(x)
#Task 4: Find and print the position of the word jones in the string
x = message.index("Jones")
print(x)
```

11/19/24- While loops

Python
```python
#while loops

#A While Loop repeats as long as
#condition is True

#Example 1:
i = 0
while i < 5 :
    print("i is:" + str(i))
    i = i+1

#Example 2
while True: #Forever Loop
    print("This will loop forever")
    break
```

11/7/24-Bytes & File sizes

End of Unit Exam 11/14 thursday 30 multiple choice questions

2^0=1
2^1=2
2^3=8
2^4=16
2^5=32

2^6=64
2^2=128

1. Alice has 600 MB of data. Bob has 2000 MB of data. Will it all fit on Alice's 4 GB thumb drive?

Yes, because the total amount of data is 2,600 GBand 1,000 MB make 1 GB and Alice has 4 GB which is 4,000MB - 2,600MB would be 1,400 MB left

2. Alice has 100 small images, each of which is 500 KB. How much space do they take up overall in MB?

   50,000MB

3. Your ghost hunting group is recording the sound inside a haunted classroom for 20 hours as MP3 audio files. How much data will that be, expressed in GB?

Audio 1MB per minute
20 hours x 60 minutes
1,200MB is 1.2GB

Here are a few more.

1. A salesperson is trying to sell you a phone that has 16 GB of memory saying, “that’s enough space to record an hour of high quality video!” This salesperson is probably wrong, but in which direction? Would you have more than enough memory or not enough?

   Yes, it would be enough because 1080p video quality is 4G to 8G per hour.

2. Shakespeare’s complete works have approximately 3.5 million characters. Which is bigger in file size: Shakespeare’s complete works stored in plain ASCII text or a 4 minute song on mp3? How much bigger?

Audio 1MB Per minute
4 minute song= 4MB
1MBx4MB=4MB
1 MB= 1 million bytes
3.5MB
The song is 0.5Mb Bigger than Shakespeare’s complete works of 3.5 million characters.

11/1/24- Binary Number system

- One= True
- zero=False
- More wires hold more bits
- Binary number system
- 0 and 1
- Decimal multiplied by the power of 10
- Binary multiplies by the power of 2
- Hexadecimal multiplied by the power of 16
- All number systems have the same rules
- More wires can store more numbers
- 32 wires can store zero to 4 billions
- 00010010= 18
- 0001 1111= 21

10/30/24- Encryption

AP EXAM-
Symmetric Encryption: Same Key used for encrypting and decrypting data.
Asymmetric Encryption(Public Key Encryption): Encrypting and decrypting data with two keys.
(Email uses asymmetric encryption) Only the user can decrypt the encrypted message.

10/28/24 Cryptography

- Encryption: The process of scrambling data into an unreadable, encoded version that only authorized users can access.
- Decryption: The process of converting an encrypted message back into the original form.
- Cipher: The algorithm used to encrypt or decrypt a message
- Key: A piece of information that unlocks the algorithm for encoding or decoding a message.
- Crack: Unscrambling an encrypted message without knowing the key.

10/24/24 Variable Swap

Python
```python
#Variable Swap

#Swap the value of x and y without using any numbers using three
lines
x=5
y=4
z=0
# Your code goes here...

z= x
x= y
y= z

print(x) # Should print 4
print(y) # Should print 5

4
5
```

10/15/24 Boolean Practice

Python
```python
#Name
#Date
#Part 1: Boolean

#(Replace the comparison operator with another that makes the
statement True.)
```

```
print(10 > 5)
print(3 >2)
print("dog" !="cat")
print(8 ==8)
print(15 < 20)
print(4 > 3)
print(2 + 2 == 4)
print(True != False)
print(not (4 > 6) == True)
print(len("python") < 7)
#Part 2: Logic
#Complete each function one at a time and test them using the
examples provided.


#Logic 1 (AND Operator)
#num = integer
def is_in_range(num):
    # Check if the number is between 10 and 20 (inclusive)
    if 10>= num and num <= 20:
        print(True)
    # Example: is_in_range(15) should print True, is_in_range(25)
should print False

is_in_range(14)

#Logic 2 (OR Operator)
#day = string
def is_weekend(day):
    # Check if the day is either Saturday or Sunday
    # Example: is_weekend("Saturday") should print True,
is_weekend("Monday") should print False
    if day == "Saturday" or day== "Sunday":
        print(True)
    else:
        print(False)

is_weekend("Saturday")
#Logic 3 (NOT Operator)
# num = integer
def is_not_negative(num):
    # Check if a number is not negative
    if not num < 0:
        print(True)
    else:
        print(False)
# Example: is_not_negative(5) should print True, is_not_negative(-3)
should print False

is_not_negative(5)
```

10/10/24

Comparison Operators

> Greater Than

< Less Than

== Equal to

> = Greater than or equal to

! = Not equal to

Logical Operators
and BOTH statements must be true
or ONE of the statements must be true
If
else
eilf

Python
```python
#conditionals

#c=init
#Functions

#Print the largest of the 3 numbers
def max_num(a,b,c):
    #No input needed
    #Process the data with conditional statements
    if a>b and a>c:
        print("a is the largest. the value of a is:" + str(a))



def score_to_grade(score):
    # No input needed
    if score> 89:
        print("A")
    elif score >79:
        print("B")
    elif score > 69:
        print("C")
    elif score > 59:
        print("D")
    else:
        print("F")

#Main
score_to_grade(50)

#Comparison Operators
#> Greater Than
#< Less Than
#== Equal to
#> = Greater than or equal to
#! = Not equal to


 #Logical Operators
 #and BOTH statements must be true
 #or ONE of the statements must be true






#conditionals

#Init
#Functions

#16 year or older
#Passed your drivers ed exam
def drive_check():
    age = int(input("Please enter your age:"))
    exam = input("Did you pass your drivers exam?") # Input by
default concatenation
```

```
    # Process w/ conditional statements
    if age > 15 and exam =="yes": #Evaluate as True or False
        print("You are eligible to obtain your license!")
    else:
        print("You are NOT eligible to obtain a license!")

#Main
drive_check()
```

10/8/24 String concatenation

String Concatenation: Linking strings together

Python
```python
# Heading

#inint
#Functions

# Main

#String Concatenation
#Linking strings together
firstName="Daniel"
lastName="Brookins"
age= 17
print(firstName+" "+lastName)
print("I am " + str(age) + " years old")
```

Python

```python
# Heading

#inint
#Functions
#This function adds two numbers together and prints the result
def sum(num1,num2):
    print(num1+num2)



#This function takes a name and prints a welcome message

def welcome(firstName):
    print("welcome to Jones,"+firstName)

# Main
welcome("Daniel Brookins")
```

9/19/24 Parameters

Step 1:Create your parameter

```
Step 2: Provide an argument to your calls
Step 3: Use your parameter in the function
```

Python

```python
ters
#A variable (container) that customizes
#The way a function works
#The value inside the parenthesis

#inint
import turtle
daniel=turtle.Turtle()
#functions
def square(size):#Step 1:Create your parameter
    for i in range(4):
        daniel.forward(100)# Step 3: Use tour parameter in the function
        daniel.left(90)
#Main
square(400)#Step 2: Provide an argument to your calls
import turtle
daniel=turtle.Turtle()
import random
#Function

import random
import turtle
daniel=turtle.Turtle()
#functions
def triangle():
    size_triangle=random.randint(0,300)
    for i in range(4):#1 triangle
        daniel.forward(size_triangle)
        daniel.right(120)
        daniel.forward(200)
    daniel.end_fill()
#Main

for i in range(10):
    triangle()
    triangle()
```

9/16/24 - Random integers

Python

Python

```python
#07 Line Project

#Initialize
import turtle
felix = turtle.Turtle()
felix.speed(1000)

daniel=turtle.Turtle()
daniel.speed(1000)
#Functions
def body(): #This function creates the body of the frog
    daniel.left(55)
    daniel.forward(200)
    daniel.left(125)
    daniel.forward(210)
    daniel.left(120)
    daniel.forward(200)
    daniel.penup()
    daniel.home()
    daniel.pendown()
def arms(): #This function creates the arms of the frog
```

```
    daniel.left(55)
    daniel.forward(100)
    daniel.penup()
    daniel.left(125)
    daniel.forward(20)
    daniel.penup()
    daniel.home()
    daniel.pendown()

def eyes(): #This function creates the eyes with the pupil in the
middle
    felix.penup()
    felix.left(90)
    felix.forward(165)
    felix.left(90)
    felix.forward(45)
    felix.left(180)
    felix.pendown()
    for i in range(2):
        felix.circle(50)
        felix.left(90)
        felix.penup()
        felix.forward(50)
        felix.right(90)
        felix.forward(25)
        felix.left(90)
        felix.pendown()
        felix.pencolor("#000000")
        felix.begin_fill()
        felix.circle(25)
        felix.end_fill()
        felix.penup()
        felix.right(125)
        felix.forward(75)
        felix.pendown()
    felix.penup()
    felix.home()
    felix.pendown()
def smile(): #This function creates the smile with the
semi-circle
    felix.penup()
    felix.left(90)
    felix.forward(150)
    felix.left(90)
    felix.forward(45)
    felix.left(180)
    felix.right(90)
    felix.pendown()
    felix.circle(50, 180)
    felix.penup()

    felix.home()
    felix.pendown()
def legs(): #This function creates the legs on both sides
utilizing semi-circles and lines
    for i in range(2):
        felix.circle(150, 75, 180)
        felix.left(105)
        felix.circle(150, 75, 180)
        felix.right(155)
    for i in range(2):
        felix.home()
        felix.left(38)
        felix.forward(150)
        felix.left(180)
        felix.forward(150)
        felix.left(280)
        felix.forward(150)
    felix.penup()
    felix.home()
    felix.pendown()
```

```
def feet(): #This function creates the feet on both sides with
the half diamonds at their ends
    for i in range(2):
        felix.right(2)
        felix.forward(150)
        felix.left(38)
        for i in range(3):
            felix.forward(20)
            felix.left(135)
            felix.forward(20)
            felix.left(180)
            felix.left(45)
        felix.left(159)
        felix.forward(150)
        felix.home()
        felix.right(195)
    felix.penup()
    felix.home()
    felix.pendown()

def frog(): #This function is creating the full frog
    body()
    feet()
    legs()
    arms()
    smile()
    eyes()

#Main
frog()

import turtle
daniel=turtle.Turtle()
daniel.forward(100)
daniel.left(180)
daniel.forward(50)
daniel.right(90)
daniel.forward(50)
daniel.left(180)
daniel.forward(100)
daniel.left(180)
daniel.forward(20)
daniel.left(90)
daniel.forward(20)
daniel.left(180)
daniel.forward(40)
daniel.left(180)
daniel.forward(20)
#Functions
#Define a new function that doesn't exist
```

9/6/24- Functions

Parameters: Size- an integer =1 (if given)
Color- Color string or a numeric color tuple

```python
#Line Project
#initialize
import turtle
daniel=turtle.Turtle()
#Function
def triangle():
    for i in range(3):
        daniel.right(120)
        daniel.forward(200)
# Make function for arms
```

```
def arms():
    for i in range(2):
        daniel.right(90)
        daniel.penup()
        daniel.forward(84)

    daniel.left(90)
    daniel.forward(50)


#Main
triangle()
arms()
```

```
import turtle
daniel=turtle.Turtle()
#Functions
#Create a function that makes a snowman
def snowman():
    daniel.dot(1000,"#ffff00")
    daniel.penup()
    daniel.pu()
    daniel.up()
    daniel.left(90)
    daniel.forward(100)
    daniel.right(90)
    daniel.forward(100)
    daniel.dot(100,"#000000")
    daniel.left(180)
    daniel.forward(250)
    daniel.dot(100,"#000000")
    daniel.forward(70)
    daniel.left(90)
    daniel.forward(200)
    daniel.penup()
    daniel.pendown()
    daniel.pensize(10)
    daniel.pencolor("#000000")
    daniel.circle(200,180)
#Main

snowman()
```

```python
#Heading

#Init
import turtle
daniel=turtle.Turtle()
#Functions
#Create a function that makes a snowman
def snowman():

    daniel.dot(120,"blue")
    daniel.left(90)
    daniel.penup()
    daniel.pu()
    daniel.up()
    daniel.forward(70)
    daniel.dot(100,"aqua")
    daniel.forward(70)
    daniel.dot(50,"light blue")
    daniel.penup()
    daniel.pu()
    daniel.up()
#Main

snowman()

#Heading

#Init
import turtle
daniel=turtle.Turtle()
#Functions
```

```python
#Create a function that makes a snowman
def snowman():
    daniel.dot(1000,"#ffff00")
    daniel.penup()
    daniel.pu()
    daniel.up()
    daniel.left(90)
    daniel.forward(100)
    daniel.right(90)
    daniel.forward(100)
    daniel.dot(100,"#000000")
    daniel.left(180)
    daniel.forward(250)
    daniel.dot(100,"#000000")
#Main

snowman()
```

9/4/24- CTF challenge fixme/Code Stutructure

def- Defines function

Python
```python
# Code Structure
#Heading

#Initialization
import turtle
daniel=turtle.Turtle()
#Functions
#Define a new function that doesn't exist
def square():
    for i in range(4):
        daniel.forward(100)
        daniel.left(90)

#Main
```

```python
#Square
#Using a function is called "Call"
square()
Square()
Square()

# Code Structure
#Heading

#Initialization
import turtle
daniel=turtle.Turtle()
daniel.left(90)
#Functions
#Define a new function that doesn't exist
def drawStep():
    daniel.forward(20)
    daniel.right(90)
    daniel.forward(20)
    daniel.left(90)

def drawSide():
    drawStep()
    drawStep()
    drawStep()
    daniel.forward(20)
    daniel.left(90)

def drawDiamond():
    drawSide()
    drawSide()
    drawSide()
    drawSide()
#Main
#Square
#Using a function is called "Call"
drawDiamond()
```

```
Python
import random



def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for
(secret_c,new_key_c) in zip(secret,new_key)])


flag_enc = chr(0x15) + chr(0x07) + chr(0x08) + chr(0x06) + chr(0x27)
+ chr(0x21) + chr(0x23) + chr(0x15) + chr(0x5a) + chr(0x07) +
chr(0x00) + chr(0x46) + chr(0x0b) + chr(0x1a) + chr(0x5a) +
chr(0x1d) + chr(0x1d) + chr(0x2a) + chr(0x06) + chr(0x1c) +
chr(0x5a) + chr(0x5c) + chr(0x55) + chr(0x40) + chr(0x3a) +
chr(0x5f) + chr(0x53) + chr(0x5b) + chr(0x57) + chr(0x41) +
chr(0x57) + chr(0x08) + chr(0x5c) + chr(0x14)


flag = str_xor(flag_enc, 'enkidu')
  print('That is correct! Here\'s your flag: ' + flag)



picoCTF{1nd3nt1ty_cr1515_182342f7}
```

8/30/24-Data Types

Integers: Whole Numbers
String: A sequence of characters inside of quotes

Variables have to be created first before variable
Variables should be descriptive and simple
Variables can't be started with numbers
Loop: Used to repeat commands

```python
#Data Types

#integers: Whole Numbers
print(100)

#string: A sequence characters
# inside of quotes
print("Hello,World12!@#%")

#Data Types

#integers: Whole Numbers
print(100)

#string: A sequence characters
# inside of quotes
#print("Hello,World12!@#%")

#Variables
#A Variable is a container or a box that you can store data in
age= 17
age= age + 1
print(age)

#Data Types

#integers: Whole Numbers
#print(100)

#string: A sequence characters
# inside of quotes
#print("Hello,World12!@#%")

#Variables
#A Variable is a container or a box that you can store data in
#name="daniel"
#print("Welcome to my website" + first)

#For Loops
```

```python
#Loop: A Loop is used to repeat a sequence of commands

import turtle
daniel=turtle.Turtle()

#Square
for i in range(4):
    daniel.forward(100)
    daniel.left(90)

#Data Types

#integers: Whole Numbers
#print(100)

#string: A sequence characters
# inside of quotes
#print("Hello,World12!@#%")

#Variables
#A Variable is a container or a box that you can store data in
#name="daniel"
#print("Welcome to my website" + first)

#For Loops
#Loop: A Loop is used to repeat a sequence of commands

import turtle
daniel=turtle.Turtle()

#Square
for i in range(3):
    print(i)


#Shapes
import turtle
daniel= turtle.Turtle()
for i in range(1):
    daniel.left(90)
    daniel.forward(200)
    daniel.left(180)
    daniel.forward(200)
    daniel.right(90)
    daniel.forward(50)
    daniel.right(90)
    daniel.forward(50)
    daniel.right(90)
    daniel.forward(50)

import turtle

x= turtle.Turtle()
x.forward(100)
x.left(90)
x.forward(200)
x.left(180)
```

8/28/24-Python Shapes

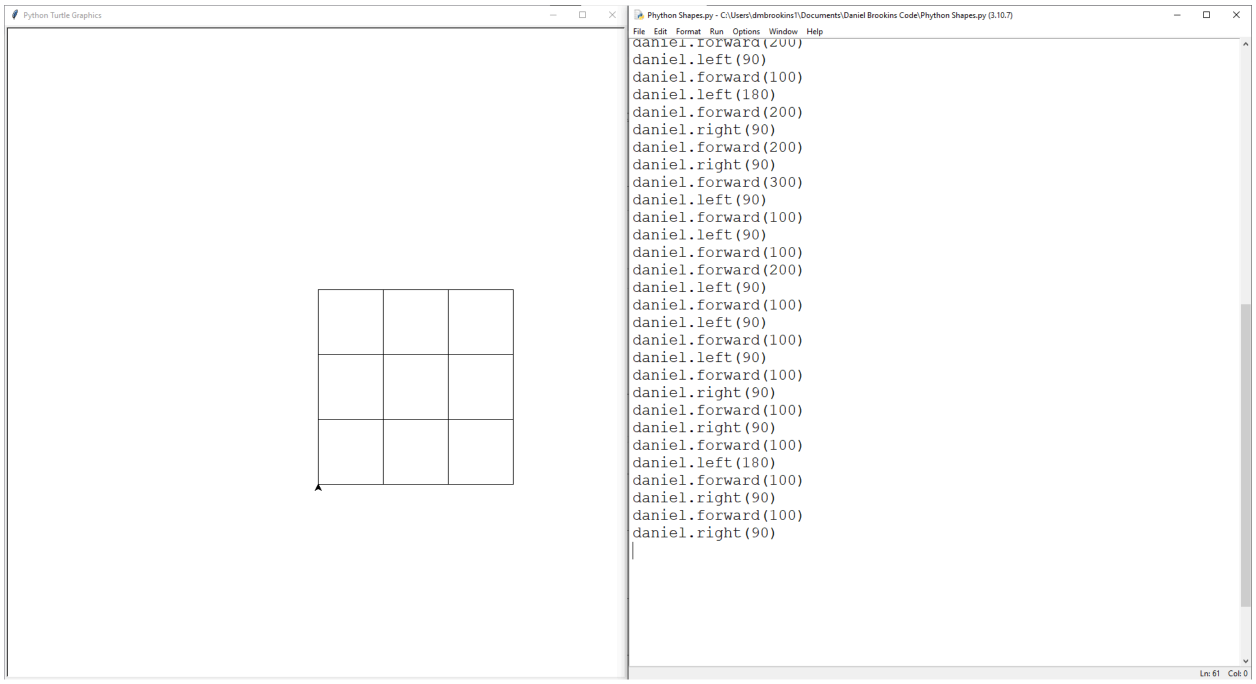
Library
Import turtle
Creating a turtle
#Daniel Brookins
#8/28/24
#Python Shapes
#We need to import a library
import turtle

#This line creates a turtle and stores it inside of a
#variable(box) called your name
daniel =turtle.Turtle()

Python

```python
daniel =turtle.Turtle()

daniel.forward(100)# Moves turtle forward
daniel.left(90)#Turns the turtle
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.right(90)
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.forward(100)
daniel.left(180)
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(180)
daniel.forward(200)
daniel.right(90)
daniel.forward(200)
daniel.right(90)
daniel.forward(300)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.left(180)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.right(90)
```


Python Turtle Graphics
Phython Shapes.py - C:\Users\dmbrookins1\Documents\Daniel Brookins Code\Phython Shapes.py (3.10.7)
File Edit Format Run Options Window Help
Ln: 61 Col: 0

```
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(180)
daniel.forward(200)
daniel.right(90)
daniel.forward(200)
daniel.right(90)
daniel.forward(300)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.forward(200)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.left(90)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.left(180)
daniel.forward(100)
daniel.right(90)
daniel.forward(100)
daniel.right(90)
```

8/26/24 -Lesson 1: Hello, World

#Daniel Brookins
#Period 3
#8/26/2024

#Task 1:Print the phrase Hello,World
Vocabulary:

Shell: A place where the output of a computer code goes and how the computer communicates any bugs or errors.

Bug: An unexpected problem with computer code

Function:print("Hello,World")

Comment: Text the computer can't read#Text that begins with a hashtag is a comment

Function:

```python
print("Hello,World")
```

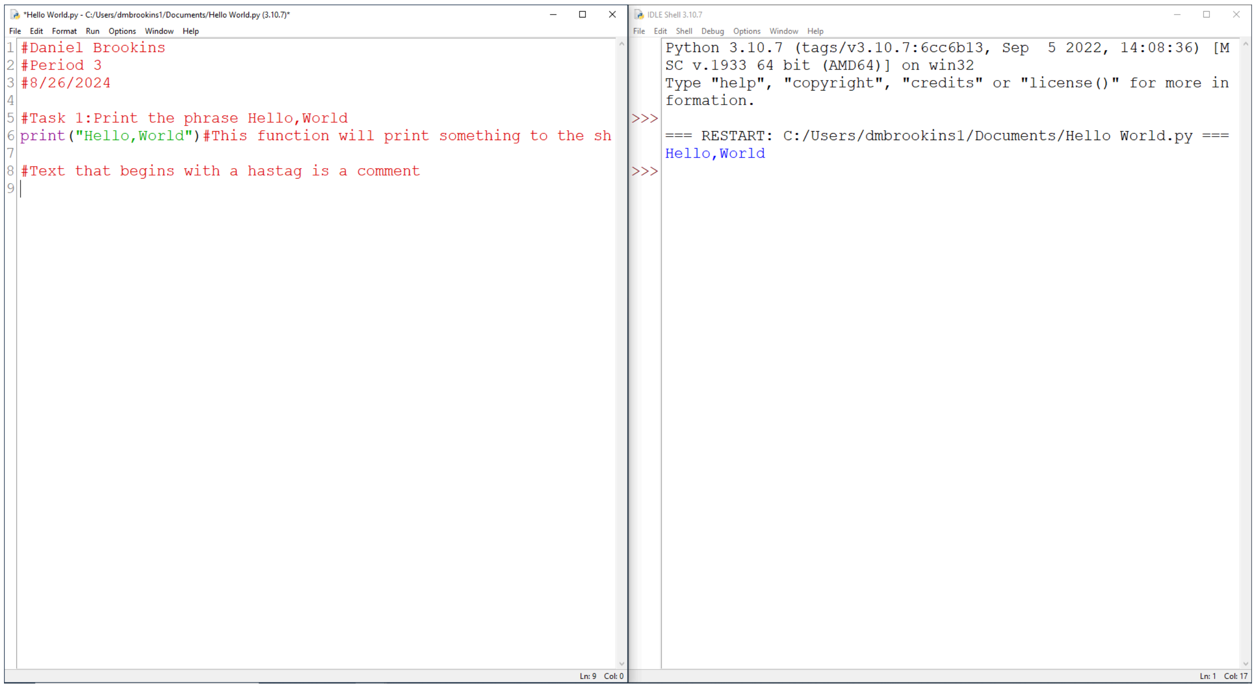
#This function will print something to the shell

Bug:(Hello,World)

Correct: ("Hello,World")

Needs "Quotation Marks define phrases"

("Hello,World") Input ("Hello,World") Output


Hello World.py - C:/Users/dmbrookins1/Documents/Hello World.py (3.10.7)

```
#Daniel Brookins
#Period 3
#8/26/2024

#Task 1:Print the phrase Hello,World
print("Hello,World")#This function will print something to the sh

#Text that begins with a hastag is a comment

```

IDLE Shell 3.10.7

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [M
SC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>> 
=== RESTART: C:/Users/dmbrookins1/Documents/Hello World.py ===
Hello,World
>>> 
```