Activity 2:

1.





2.

A number with numbers on it

Description automatically generated with medium confidence

A close up of numbers

Description automatically generated

3.

A screenshot of a computer code

Description automatically generated

****

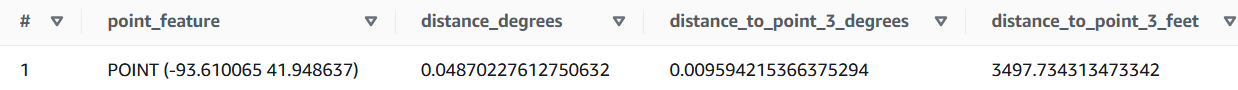
4.

A computer screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated



5.

A screenshot of a computer

Description automatically generated

A computer screen with text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

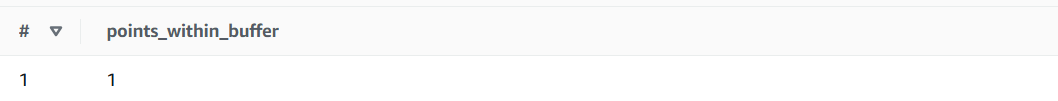
6.

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer code

Description automatically generated



Activity 3:

A close-up of a computer screen

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

Activity 4:

A screenshot of a computer

Description automatically generated

import json

import boto3

import os

s3\_client = boto3.client('s3')

def lambda\_handler(event, context):

# Log the received event

print("Received event: " + json.dumps(event, indent=2))

# Extract the bucket name and key from the event

for record in event['Records']:

source\_bucket = record['s3']['bucket']['name']

source\_key = record['s3']['object']['key']

# Ensure we're processing files from the correct source bucket

if source\_bucket == 'reactorlab':

destination\_bucket = 'reactor-lab-output'

# Copy the file from the source bucket to the destination bucket

copy\_source = {

'Bucket': source\_bucket,

'Key': source\_key

}

try:

# Perform the copy operation

s3\_client.copy\_object(

CopySource=copy\_source,

Bucket=destination\_bucket,

Key=source\_key

)

# Optionally delete the file from the source bucket after copying

s3\_client.delete\_object(Bucket=source\_bucket, Key=source\_key)

print(f"Successfully moved {source\_key} from {source\_bucket} to {destination\_bucket}")

except Exception as e:

print(f"Error occurred while moving {source\_key}: {str(e)}")

return {

'statusCode': 200,

'body': json.dumps('Lambda function executed successfully!')

}

Activity 5: