**Nikita Bajracharya**

**Assignment 1**

**19 September 2019**

1. Create three variables in a single line and assign different values to them and make sure their data types are different. Like one is int, another one is float and last one is string.

x,y,z = 5 , "Nikita", 3.8

print (type(x))

print (type(y))

print (type(z))

1. Create a variable of value type complex and swap it with another variable whose value is an integer.

comp\_x = 1+3j

int\_y = 3

# Swapping two numbers using temporary variable

temp = comp\_x

comp\_x = int\_y

int\_y = temp

print("Value of comp\_x after swapping : ", comp\_x)

print("Value of int\_y after swapping : ", int\_y)

1. Swap two numbers using third variable as result name and do the same task without using any third variable.

x = 100

y = 25.25

# Swapping two numbers using third variable named result variable

result\_name = x

x = y

y = result\_name

print("Value of x after swapping : ", x)

print("Value of y after swapping : ", y)

X = 100

y = 25.25

# Swapping two number without using third variable

x,y = y,x

print("Value of x after swapping : ", x)

print("Value of y after swapping : ", y)

1. Write a program to print the value given by the user by using both Python 2.x and Python 3.x Version.

>>> num = raw\_input("Enter any value :")

Enter any value :25

>>> print num

25

num = input("Enter any value : ")

print(num)

1. Write a program to complete the task given below:

* Ask user to enter any 2 numbers in between 1-10 and add both of them to another variable call z.
* Use z for adding 30 into it and print the final result by using variable result.

x = input("Enter your first value (1-10) : ")

y = input("Enter your second value (1-10) : ")

print(x)

print(y)

#print z which is sum of x and y

z = int(x) + int(y)

print(z)

#add 30 to z

z = z + 30

print(z)

1. Write a program to check the data type of the entered values. HINT: Printed output should say - The input value data type is : int/float/string/etc

value = input("Enter your value : ")

print(value)

print("The value is of type : ", type(value))

1. Create Variable using CamelCase, LadderCase and UPPERCASE

firstName = "Nikita" #camelcase

LAST\_NAME = "Bajracharya" #UpperCase

full\_name = firstName + LAST\_NAME

print(full\_name)

**FEEDBACK :**

In question no 5: You can also ask the user to input only integer values by providing

int(input(“ “)) rather than doing z= int(x) + int(y)

Rest all the questions are correct. Great work!

**Assignment 2**

**20 September 2019**

8. If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value. If Yes then Why?

Yes, the value assigned to variable ‘a’ will change because the initial value of ‘a’ will be lost once we reassigned different data type value, and it will remain in the memory until we assign different values.

**TASK TWO: OPERATORS AND DECISION MAKING STATEMENT**

1. Write a program in Python to perform the following operation:

· If a number is divisible by 3 it should print “Consultadd” as a string

· If a number is divisible by 5 it should print “Python Training” as a string

· If a number is divisible by both 3 and 5 it should print “Consultadd Python Training” as a string.

· #If a number is divisible by 3 it should print “Consultadd” as a string

· num = int(input("Enter a number : "))

· if num/3:

· print("Consultadd")

·

· #If a number is divisible by 5 it should print “Python Training” as a string

· num = int(input("Enter a number : "))

· if num/5:

· print("Python Training")

·

· #If a number is divisible by both 3 and 5 it should print “Consultadd Python Training” as a string

· num = int(input("Enter a number : "))

· if num/3 and num/5:

· print("Consultadd Python Training")

2. Write a program in Python to perform the following operator based task:

· Ask user to choose the following option first:

o If User Enter 1 - Addition

o If User Enter 2 - Subtraction

o If User Enter 3 - Division

o If USer Enter 4 - Multiplication

o If User Enter 5 - Average

· Ask user to enter the 2 numbers in a variable for first and second for the first 4 options mentioned above.

· Ask user to enter two more numbers as first1 and second2 for calculating the average as soon as user choose an option 5.

· At the end if the answer of any operation is Negative print a statement saying “Oops option X(1/2/3/4/5/) is returning the negative number”

· NOTE: At a time user can perform one action at a time.

print("1 - Addition")

print("2 - Substraction")

print("3 - Division")

print("4 - Multiplication")

print("5 - Division")

#Ask user to choose option from 1-5

option = input("Enter option(1/2/3/4/5) : ")

#Ask user to enter the 2 numbers

first = int(input("Enter first number: "))

second = int(input("Enter second number: "))

if option == '1':

output = first + second

print ("Addition :", output)

elif option == '2':

output = first - second

print("Substraction :", output)

elif option == '3':

output = first / second

print("Division :", output)

elif option == '4':

output = first \* second

print("Multiplication :",output)

elif option == '5':

first1 =int(input("Enter first number for average task :"))

second2 = int(input("Enter second number for average task :"))

output = (first + second + first1 + second2)/4

print("Average :", output )

else:

print("No Valid Option")

#Check if output if negative

if output < 0:

print("Oops option X(1/2/3/4/5/) is returning the negative number")

3. Write a program in Python to implement the given flowchart:

a = 10

b = 20

c = 30

#calculate average of three variable and print it

avg = (a+b+c)/3

print("avg =", avg)

if (avg > a) and (avg > b) and (avg > c):

print("avg ig higher than a,b,c")

elif (avg > a) and (avg > b):

print("avg is higher than a,b,c")

elif (avg > a) and (avg > c):

print("avg is higher than a, c")

elif (avg > b) and (avg > c):

print("avg is higher than b,c")

elif (avg > a):

print("avg is just higher than a")

elif (avg > b):

print("avg is higher than b")

elif (avg > c):

print("avg is just higher than c")

4. Write a program in Python to break and continue if the following cases occurs:

● If user enters a negative number just break the loop and print “It’s Over”

● If user enters a positive number just continue in the loop and print “Good Going”

#break and continue task

num = int(input("Enter a number : "))

while num > 0:

break

print("It's Over")

while num <= 0:

continue

print("Good Going")

5. Write a program in Python which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200.

#Find all numbers divisible by 7 and are not multiple of 5

for num in range(2000, 3200):

if (num % 7 == 0 and num % 5 != 0):

print(num)

6. What is the output of the following code examples?

● x=123

for i in x:

print(i)

**Output :** TypeError: 'int' object is not iterable

● i = 0

while i < 5:

print(i)

i += 1

if i == 3:

break

else:

print(“error”)

**Output :**

0

error

1

error

2

● count = 0

while True:

print(count)

count += 1

if count >= 5:

Break

**Output:**

0

1

2

3

4

7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.

Expected output: 0 1 2 4 5

Note: Use ‘continue’ statement

for num in range(6):

if num == 3 or num == 6:

continue

print(num,end=" ")

8. Write a program that accepts a string as an input from user and calculate the number of digits and letters.

Expected output: consul12

Letters 6

Digits 2

#A program that accepts a string as an input from user

#and calculate the number of digits and letters

string = input("Enter a string: ")

digit\_counter = 0

letter\_counter = 0

for i in string:

if i.isdigit():

digit\_counter += 1

elif i.isalpha():

letter\_counter += 1

else:

pass

print ("Letters ", letter\_counter)

print("Digits ", digit\_counter)

9. Read the two parts of the question below:

● Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.

● Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. ( The program continues as long as a user has not answered “no” and has not guessed the correct number)

#Guess the lucky number task

guess\_number = input("Guess the lucky number :")

while guess\_number != 5:

print("Wrong guess")

guess\_number = input("Guess the lucky number :")

#guess number again each time

guess\_number = -1

answer = "yes"

while guess\_number != 5 and answer != "no":

guess\_number = input("Guess the lucky number :")

if guess\_number != 5:

print("Wrong guess")

answer = input("Would you like to guess again?")

10 . Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as

counter=1

While counter <= 5:

print(“Type in the”, counter, “number”

counter=counter+1

The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess it stops and prints “Game over!”.

#Program that asks 5 times to guess the lucky number

counter = 1

while counter <= 5:

guess\_number = input("Guess the "+ str(counter)+ " number :")

counter = counter + 1

if guess\_number != 5:

print("Try again!")

else:

print("Good guess!")

else:

print("Game over!")

11. In the previous question, insert “break” after the “Good guess!” print statement. “break” will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.

#Program that asks 5 times to guess the lucky number and break after "Good Guess"

counter = 1

while counter <= 5:

guess\_number = input("Guess the "+ str(counter)+ " number :")

counter = counter + 1

if guess\_number != 5:

print("Try again!")

else:

print("Good guess!")

break

else:

print("Sorry but that was not very successful")

**FEEDBACK:**

In the question no 10 and 11, The code is not giving the right output even after providing the right guess no. Check it again.

**Assignment 3**

**24 September 2019**

**TASK THREE: DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

#create a list of 10 elements of 4 different data types

list = [2, 3.56, "BOB", 1+5j, "Ryan", 3j, 9.20, 5, 2+8j, "Seema"]

print(list)

2. Create a list of size 5 and execute the slicing structure.

#create a list of size 5 and execute the slicing structure

list = [2, 3, 4, 5, 6]

print(list)

print(len(list)) #length

print(list[1:4]) #Range

print(list[2:]) #omit the end point of the slice and print the end part

print(list[:5]) #omit the start point of the slice and start from the beginning

print(list[-4:-2]) #negative indexing of range

print(list[-3:]) #make a slice of last 3 elements

print(list[::2]) #print a slce of every other element

3. Write a program to get the sum and multiply of all the items in a given list.

#sum and multiple of all the items in a given list

list = [2, 0, 6, 4, 9]

sum = 0

multiple = 1

for x in list:

sum = sum + x

multiple = multiple \* x

print("Sum og all the items in List : ", sum)

print("Multiple of all items in List :", multiple)

4. Find the largest and smallest number from a given list.

list = [2, 0, 6, 4, 9]

print("Largest item in the List : ", max(list)) >>>9

print("Smalles item in the List : ", min(list)) >>>

#largest and smallest number in the given list

grade\_list = []

size = int(input("Please enter the length for grade\_list :"))

for x in range(1, size+1):

grade = int(input("Please enter the grade : "))

grade\_list.append(grade)

print("Highest grade : ", max(grade\_list))

print("Lowest grade : ", min(grade\_list))

5. Create a new list which contains the specified numbers after removing the even numbers from a predefined list.

#Print new list after removing even number

List = [1,2,3,4,5,6,7,8,9,10]

print("Old List :", List)

for x in List:

if x%2 == 0:

List.remove(x)

print("New List : ", List)

6. Create a list of first and last 5 elements where the values are square of numbers between 1 and 30 (both included).

#Print square of numbers between 1 and 30

List = []

for x in range(1,21):

List.append(x\*\*2)

print(List[:5])

print(List[-5:])

7. Write a program to replace the last element in a list with another list.

Sample data: [1,3,5,7,9,10],[2,4,6,8]

Expected output: [1,3,5,7,9,2,4,6,8]

#Program to replace the last element of list with another list

List1 = [1,3,5,7,9,10]

List2 = [2,4,6,8]

List1[-1:] = List2

print(List1)

8. Create a new dictionary by concatenating the following two dictionaries:

a={1:10,2:20}

b={3:30,4:40}

Expected Result: {1:10,2:20,3:30,4:40}

#Program to create new dictionary by concatenating the two dictionaries

a = {1:10,2:20}

b = {3:30,4:40}

c = {}

for x in (a,b):

c.update(x)

print(c)

9. Create a dictionary that contains a number (between 1 and n) in the form(x,x\*x).

Sample data (n=5)

Expected Output: {1:1,2:4,3:9,4:16,5:25}

#Program to create a dictionary that contains a numb(1 and n) in the form (x,x\*x)

n = int(input("Enter the sample number : "))

dictionary = {}

for x in range(1, n+1):

dictionary[x] = x\*x

print (dictionary)

10. 10. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

Suppose the following input is supplied to the program:

34,67,55,33,12,98

The output should be:

[‘34’,’67’,’55’,’33’,’12’,’98’]

(‘34’,’67’,’55’,’33’,’12’,’98’)

#Program to create list and tuples from comma separated numbers

number = input("Input some comma separated numbers : ")

list = number.split(",")

tuple = tuple(list)

print(list)

print(tuple)

**FEEDBACK:**

In question no 5 , you can also create an empty list and then add items in that new list. Try with this approach as well.

Great Work!

**Nikita Bajracharya**

**Assignment 4**

**Data Structure, Functions and Higher Order Function**

**WEEKEND ACTIVITY ON DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

#create a list of 10 elements of 4 different data types

List = [2, 3.56, "BOB", 1+5j, "Ryan", 3j, 9.20, 5, 2+8j, "Seema"]

print(List)

2. Create a list of size 5 and execute the slicing structure

#create a list of size 5 and execute the slicing structure

List = [2, 3, 4, 5, 6]

print(List)

print(len(List)) #length

print(List[1:4]) #Range

print(List[2:]) #omit the end point of the slice and print the end part

print(List[:5]) #omit the start point of the slice and start from the beginning

print(List[-4:-2]) #negative indexing of range

print(List[-3:]) #make a slice of last 3 elements

print(List[::2]) #print a slce of every other element

3. Create a list of given structure and run

x= [100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

· Access list [1, 2, 3, 4]

· Access list [600, 700]

· Access list [100, 300, 500, 600, 800]

· Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]

· Access list [10]

· Access list [ ]

4. Create a list of thousand number using range and xrange and see the difference between each other.

#Create a program using range and xrange for 1000 numbers

x = range(1,1000) #prints a list from 1 to 1000

print(list(x))

x1 = xrange(1,1000) #works only in python 2

print(list(x1))

5. How Tuple is beneficial as compare to the list?

Apart from being an immutable data structure, Tuples operation are faster than lists which can be measure using the ‘timeit’ library. Some tuples can be used as dictionary keys but lists can never be used as dictionary keys because they are not immutable. Tuples can be used as values in sets whereas lists can not.

6. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

#Program to iterate through the list of nums in the range 1,100 and print the number which is divisible by 3 and a multiple of 2

list = []

for i in range(1100):

if (i % 3 == 0) and (i%2 == 0):

list.append(str(i))

print(",".join(list))

7. Write a program in Python to reverse a string and print only the vowel alphabet if exist in the string with their index.

#Program to reverse a string and print only the vowel alphabet

#if exist in the string with their index

def reverse\_string(s):

return s[::-1]

vowel = ['a','e','i','o', 'u', 'A','E','I','O','U']

string = input("Please input a word :")

new\_word = reverse\_string(string)

print("Reverse string is ", new\_word)

#loop to check vowel in reversed word

for i in range(len(new\_word)):

if new\_word[i] in vowel:

print("Vowel alphabet : ", new\_word[i], " on position : ", i)

8. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which has even length of word.

#Program to iterate through the string "hello my name is abcde"

#print the string which has even length of word

string = "hello my name is abcde"

extractWords = list(string.split(' '))

#Check the single word from string

print(extractWords)

#loop to check is word has even length

for i in extractWords:

if len(i) % 2 == 0:

print("Even length words are ", i)

9. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

x=[1,2,3,4,5,6,7,8,9,-1]

#program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

def printPairs(x, n, sum):

for i in range(0,n):

for j in range (i+1,n):

if(x[i] + x[j] == sum):

print(x[i], "+" ,x[j], "=", sum)

x = [1,2,3,4,5,6,7,8,9,-1]

n = len(x)

sum = 8

printPairs(x,n,sum)

10. Write a program in Python to complete the following task:

● Create two different list as in even\_list and odd\_list

● Ask user to enter the number in the range of 1,50 and make sure if the entered number is even append it to the even\_list and if the entered number is odd append it to the odd list.

● Keep that in mind you can only add 5 items in each list

● Make sure once you entered the total 5 element calculate the sum of the list and return the maximum out of the list.

11. Write a program to find out the occurrence of a specific word from an alphanumeric statement. **Example:** 12abcbacbaba34ab

**Output:** a=5 b=5 c=2 make sure you should avoid the numbers in you logic

#Program to count a specific word from an alphanumeric statement

def frequency\_words(statement):

dict = {}

for char in statement:

keys = dict.keys()

if char in keys:

dict[char] += 1

else:

dict[char] = 1

return dict

statements =str(input("Enter an alphamumeric statememt: "))

alphaStatements = ''.join([i for i in statements if not i.isdigit()])

print(frequency\_words(alphaStatements))

**WEEKEND ACTIVITY ON FUNCTIONS**

1. Write a program to reverse a string.

Sample data: “1234abcd”

Expected Output: “dcba4321”

#Program to reverse a string

def reverse(string):

str = ""

for i in string:

str = i + str

return str

my\_string = input("Enter a string: ")

print("Original string : ", end= "")

print(my\_string)

print("Reversed string : ", end= "")

print(reverse(my\_string))

2. Write a function that accepts a string and calculate the number of upper case letters and lower case letters.

#Function that accepts a string and calculate th num of Upper and lower case letters

def case\_detection(string):

uppercase\_count = 0

lowercase\_count = 0

for i in string:

if i.isupper():

uppercase\_count += 1

elif i.islower():

lowercase\_count += 1

print("Original string: ", string)

print("No of upper case letters - ", uppercase\_count)

print("No of lower case letters - ", lowercase\_count)

case\_detection("UPPER and Lower CaSE funTions")

3. Create a function that takes a list and return a new list with unique elements of the first list.

#Functions that takes a list

#Return a new list with unique elements of the first list

def unique\_list(list1):

new\_list = []

#Traverse for all elements

for i in list1:

#Checks if elements exists in list1

if i not in new\_list:

new\_list.append(i)

else:

pass

return new\_list

list = [1,3,5,5,7,2,9,9]

print("The unique elements from old list : " )

print(unique\_list(list))

4. Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

#Program that accepts a hyphen-separated sequence of words as input

#Prints the word in a that after sorting them alphabetically

def hyphen\_program(word):

my\_words = word.split('-')

my\_words.sort()

for i in my\_words:

print ('-'.join(my\_words))

user\_word = input("Enter any hyphen(-) separated words: ")

hyphen\_program(user\_word)

5. Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Sample input:

Hello world

Practice makes perfect

Expected Output:

HELLO WORLD

PRACTICE MAKES PERFECT

#Program that takes sequence of lines as input

#Prints the lines making them capitalized

lines = []

while True:

line = input()

if len(line) == 0:

break

lines.append(line.upper())

for line in lines:

print(line)

6. Define a function that can receive two integral numbers in string form and compute their sum and print it in console.

#Funtion that can receive two integral number in the string form

#compute their sum

def computeSum(x, y):

sum = int(x) + int(y)

return sum

#Take number in string format

num1 = "19"

num2 = "27"

total\_sum = computeSum(num1, num2)

print("Sum : ", total\_sum)

7. Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print all strings line by line.

#Function that can accept two strings and print the string with maximum length

#Same length will print all strings line by line

def maxLength(x, y):

if (len(x) == len(y)):

print(x)

print(y)

elif (len(x) > len(y)):

print(x)

else:

print(y)

string1 = input("Enter 1st string: ")

string2 = input("Enter 2nd string: ")

maxLength(string1,string2)

8. Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20.

#Funtion which can generate and print a tuple where the value are square num between 1 and 20

def squareNum():

list = []

for i in range(1,20):

list.append(i\*\*2)

t = tuple(list)

print(t)

squareNum()

9. Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

Example: If the limit is 3 , it should print:

0 EVEN

1 ODD

2 EVEN

3 ODD

#Function called showNumbers and print all the numbers between

#0 and limit with a label to identify even and odd numbers

limit = int(input("Enter a limit :"))

def showNumbers(limit):

for i in range(0, limit+1):

if i%2 == 0:

print(str(i), " EVEN")

else:

print(str(i), " ODD")

showNumbers(limit)

**10.**  Write a program which can filter() to make a list whose elements are even number between 1 and 20 ( both included)

2. #Program using filter() to make a list whose elements are even between 1 and 20

3. li = [i for i in range(1,21)]

4. print(li)

5. evenNums = list(filter(lambda x: x%2==0, li))

6. print(evenNums)

11. Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]

Hints: Use map() to generate a list.

Use filter() to filter elements of a list

Use lambda to define anonymous functions

#Program to make a list whose elements are square of even number

# in [1,2,3,4,5,6,7,8,9,10]

li = [1,2,3,4,5,6,7,8,9,10]

print(li)

squareNums = list(map(lambda x: x\*\*2, filter(lambda x: x%2 == 0, li)))

print(squareNums)

12. Write a function to compute 5/0 and use try/except to catch the exceptions

#Exception Handling for 5/0 using try except

try:

x = 5/0

except:

print(" ERROR - A number cannot be divided by zero")

13. Flatten the list [[1,2,3].,[4,5],[6,7,8]] into [1,2,3,4,5,6,7,8] using reduce

Goal : Turn [1,2,3,4,5,6,7] to 1234567

#Program using reduce

import functools

li = [[1,2,3],[4,5],[6,7,8]]

#newList = [j for i in li for j in i]

#print(newList)

list1 = functools.reduce(lambda x,y: x+y, li)

print(list1)

#iterating each element for conversion

for i in list1:

print(i, end="")

14. What is the output of the following codes:

* def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

**Output : 2**

* def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()

**Output: NameError: name 'f' is not defined**

**TASK FOUR: HIGHER ORDER FUNCTIONS, GENERATORS, LIST COMPREHENSION AND DECORATOR**

1. Write a program to Python find the values which is not divisible 3 but is should be a multiple of 7. Make sure to use only higher order function.

#Program that use higher order function to find the value

#not divisible by 3 but a multiple of 7

li = range(1,50)

value = list(filter(lambda x: (x%3 != 0 and x%7 == 0),li))

print(value)

>>> [7, 14, 28, 35, 49]

2. Write a program in Python to multiply the element of list by itself using traditional function and pass the function to map to complete the operation.

#Program to multiply the element of list by itself using traditional

#function and pass the function to map to complete the operation

def mult\_elements(x):

return x\*x

list1 = range(1,10)

mult\_result = map(mult\_elements, list1)

print(list(mult\_result))

>>> [1, 4, 9, 16, 25, 36, 49, 64, 81]

3. Write a program to Python find out the character in a string which is uppercase using list comprehension.

#Program to find out the character in a string which is uppercase using list comprehension

str = "PythonTraining"

upper\_letter = [x for x in str if x.isupper()]

print(upper\_letter)

>>> ['P', 'T']

4. Write a program to construct a dictionary from the two lists containing the names of students and their corresponding subjects. The dictionary should maps the students with their respective subjects. Let’s see how to do this using for loops and dictionary comprehension. HINT-Use Zip function also

* Student = ['Smit', 'Jaya', 'Rayyan']
* capital = ['CSE', 'Networking', 'Operating System']

· #Program using dictionary and map

· student = ['Smit','Jaya','Rayyan']

· capital = ['CSE','Networking', 'Operating System']

· #dictionary comprehension

· dict1 = {keys:values for (keys, values) in zip(student, capital)}

· print(dict1)

5. Learn More about Yield, next and Generators

6. Write a program in Python using generators to reverse the string. Input String = “Consultadd Training”

#Program to reverse a string using generator

def reverse\_string(s):

str = ""

index = len(s)

while index > 0:

str += s[index-1]

index = index - 1

yield str

for x in reverse\_string("Consultadd Training"):

print(x)

7. Write any example on decorators.

#Decorator extends the functionality of one of function without modifying another

#change the behavior of existing function

def div(x,y):

print(x/y)

def check\_div(func):

def inner\_func(x,y):

if y == 0:

print("Num cannot be divided by 0")

return

func(x,y)

return inner\_func

div = check\_div(div)

div(5,0)

#Decorators example- a function that wrap a function and modify

#its behavior in one way or the another without having to directly

#change the source code of the function being decorated

def my\_decorator(func):

def my\_wrapper():

print("~" \* 12)

func()

print("~" \* 12)

return my\_wrapper

@my\_decorator

def say\_hello():

print("Hello World")

#hello = my\_decorator(say\_hello)

#hello()

say\_hello()

8. **Learn about What is FRONT END and its Technologies and Tools**

* Make sure to mention at least 5 top notch technologies of Frontend.
* Also mentioned the name of companies using those 5 technologies individually

Front-end basically refers to the user interface and front end web development (client-side development) is the practice of producing HTML, CSS and JavaScript for a website or web application where a user can see and interact with them directly. If a person is a front-end developer he architects and develop website and web application using above technologies which typically runs on the Open Web Platform or acts as a input for non-web platform environments that is React Native.

1. ReactJS – Facebook, Instagram, Netflix, WhatsApp, Yahoo! Mail

2. Angular – Google

3. Grunt – Adobe, Walmart, Microsoft Corporation

4. Bootstrap – Spotify, twitter, Intel, Udemy

5. NPM – reddit, Starbucks, Yahoo! Mail

**TASK FIVE: FILE HANDLING AND EXCEPTION HANDLING**

1. Write a program in Python to allow the error of syntax to go in exception. HINT: use SyntaxError

#Program to allow the error to go in exception

try:

eval("x >== 2")

except SyntaxError:

print("There is a problem in a structure of your code")

2. Write a program in Python to allow user to open a file by using argv module. If the entered name is incorrect throw an exception and ask them to enter the name again. Make sure to use read only mode.

#Program to allow user to open a file by using argv module

#If entered name is incorrect throw can exception and ask them to enter the name again.

from sys import argv

nameofprogram, filename = argv

print("Name of the program: ", nameofprogram)

print("Name of the file: ", filename)

counter = 0

if counter <= 3:

while True:

try:

filehandler = open(filename)

content = filehandler.read()

print(content)

filehandler.close()

break

except:

print("Please enter the correct name of file: ")

input\_name = input("Would like to enter the name again? [Y/N] :")

if input\_name == "Y":

filename = input("Please enter the correct name of file: ")

else:

break

else:

sys.exit(1)

3. Write a program to handle an error if the user entered the number more than four digits it should return “Please length is too short/long !!! Please provide only four digits”

try:

x = int(input("Please enter any numbers : "))

counter = 0

while (x > 0):

x = x//10

counter = counter + 1

print(counter)

if counter == 4:

print("You entered a correct digits")

except:

print("Please length is too short/long!!!Please provide only 4 digits ")

4. Create a login page backend to ask user to enter the UserEmail and password. Make sure to ask Re-Type Password and if the password is incorrect give chance to enter it again but it should not be more than 3 times.

5. <https://www.programiz.com/python-programming/exception-handling> Go through this link to understand Finally and Raise concept.

6. Read any file using Python File handling concept and return only the even length string from the doc.txt file.

Consider the content as:

Hello I am a file

Where you need to return the data string

Which is of even length

Make sure you return the content in

The same link as it is present.

#Count even length string in a file

file = open('doc.txt','r')

numWords = 0

for line in file:

wordLists = list(line.split(' '))

numWords =+ len(wordLists)

#iterte each word, and print even word

for words in wordLists:

if(len(words)%2 == 0):

print(words)

file.close()

**TASK SIX: CLASSES AND OBJECTS**

1. Write a program that calculates and prints the value according to the given formula:

Q= Square root of [(2\*C\*D)/H]

Following are the fixed values of C and H:

C is 50. H is 30.

D is the variable whose values should be input to your program in a comma-separated sequence.

#Program that calculates and print the value

from math import \*

C,H = 50, 30

#function that returns formula

def my\_formula(D):

return sqrt((2\*C\*D)/H)

D = input().split(',')

D = [str(round(my\_formula(int(i)))) for i in D]

print(",".join(D))

2. Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape’s area is 0 by default.

# Program implementing inheritance

#class named Shape with init function and area

class Shape(object):

def \_\_init\_\_(self):

pass

def area(self):

return 0

#class named Square

class Square(Shape):

def \_\_init\_\_(self,len):

Shape.\_\_init\_\_(self)

self.length = len

def area(self):

return self.length\*self.length

secondSquare = Square(9)

print(secondSquare.area())

3. Create a class to find the three elements that sum to zero from a set of n real numbers.

Input array: [-25,-10,-7,-3,2,4,8,10]

Output: [[-10,2,8],[-7,-3,10]]

#Searching for 3 elements that sum to zero from a list of n real numbers

def find\_mysum(lst, n):

result = []

sum = True

for i in range(0, n-2):

for j in range(i+1, n-1):

for k in range(j+2, n):

if(lst[i]+lst[j]+lst[k] == 0):

result.append([lst[i],lst[j],lst[k]])

print(result)

sum = True

#Print the statement if there is no elements that sum to 0

if sum == False:

print("There are no elements that sum to 0")

#Driver Code

lst = [-25,-10,-7,-3,2,4,8,10]

n = len(lst)

find\_mysum(lst,n)

4. What is the output of the following code? Explain your answer as well.

· class Test:

def \_\_init\_\_(self):

self.x = 0

class Derived\_Test(Test):

def \_\_init\_\_(self):

self.y = 1

def main():

b = Derived\_Test()

print(b.x,b.y)

main()

**Output:** AttributeError: 'Derived\_Test' object has no attribute 'x'

There will be an error during an execution of above code because although child class “Derived\_Test” inherits a parent class “Test”, variable x wasn’t inherited.

· class A:

def \_\_init\_\_(self, x= 1):

self.x = x

class der(A):

def \_\_init\_\_(self,y = 2):

super().\_\_init\_\_()

self.y = y

def main():

obj = der()

print(obj.x, obj.y)

main()

**Output:** 1 2

super().\_\_init\_\_() in a sub class is referencing the parent class explicitly and invoke a init method of parent class A from der.

· class A:

def \_\_init\_\_(self,x):

self.x = x

def count(self,x):

self.x = self.x+1

class B(A):

def \_\_init\_\_(self, y=0):

A.\_\_init\_\_(self, 3)

self.y = y

def count(self):

self.y += 1

def main():

obj = B()

obj.count()

print(obj.x, obj.y)

main()

**Output:** 3 1

When obj.count() is called, the value of y become 1.

· class A:

def \_\_init\_\_(self):

self.multiply(15)

print(self.i)

def multiply(self, i):

self.i = 4 \* i;

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

def multiply(self, i):

self.i = 2 \* i;

obj = B()

**Output**: 30

5. Create a Time class and initialize it with hours and minutes.

Make a method addTime which should take two time object and add them. E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)

Make a method displayTime which should print the time.

Make a method DisplayMinute which should display the total minutes in the Time. E.g.- (1 hr 2 min) should display 62 minute.

#Program that display time and minutes

class Time:

def \_\_init\_\_(self, hours, minutes):

self.hours= hours

self.minutes = minutes

def addTime(time1, time2):

time3 = Time(0,0)

if time1.minutes + time2.minutes > 60:

time3.hours = (time1.minutes + time2.minutes)/60

time3.hours += time1.hours + time2.hours

time3.minutes = (time1.minutes + time2.minutes)-(((time1.minutes+time2.minutes)/60)\*60)

return time3

def displayTime(self):

print(int(self.hours), "hours and",int(self.minutes), "minutes")

def displayMinute(self):

print((int(self.hours)\*60)+int(self.minutes), "minutes")

#Driver Code

t1 = Time(2,50)

t2 = Time(1,20)

t3 = Time.addTime(t1,t2)

t3.displayTime()

t3.displayMinute()

6.Write a Person class with an instance variable, , and a constructor that takes an integer, , as a parameter. The constructor must assign to after confirming the argument passed as is not negative; if a negative argument is passed as , the constructor should set to and print Age is not valid, setting age to 0.. In addition, you must write the following instance methods:

1. yearPasses() should increase the instance variable by .

2. amIOld() should perform the following conditional actions:

o If , print You are young..

o If and , print You are a teenager..

o Otherwise, print You are old..

Sample Input:

4

-1

10

16

18

Sample Output:

Age is not valid, setting age to 0.

You are young.

You are young.

You are young.

You are a teenager.

You are a teenager.

You are old.

You are old.

You are old.

#Program that print Age of a person

class Person:

age = 0

#constructer that takes an integer

def \_\_init\_\_(self, initialAge):

#code to set age to 0 if Age is not valid

if initialAge < 0:

print("Age is not valid, and set age to 0")

else:

self.age = initialAge

def amIOld(self):

if self.age < 13:

print("You are young")

elif self.age >= 13 and self.age < 18:

print ("You are a teenager")

else:

print("You are old")

#dunction that increment the instance variable

def yearPasses(self):

self.age += 1

numOfInput = int(input())

for i in range(0, numOfInput):

age =int(input("Please input your age: "))

p1 = Person(age)

p1.amIOld()

for j in range(0,2):

p1.yearPasses()

p1.amIOld()

print(" ")

**WEEKEND ACTIVITY- THEORETICAL JOURNEY**

1. What is Pickling and Unpickling in Python? Explain with the help of example.

**Pickling** is the process whereby a Python object structure is converted into a byte stream that is a string of bytes that can be written to any file-like objects. The pickle module implements powerful algorithm for serializing and deserializing a Python object structure. It is useful when you want to save the state of objects and reuse them for another time.

**Unpickling** is the process of reconstructing the object from the byte stream.

#pickling and unplicking

#pickling pickle.dump(oject,file)

import pickle

class Employee:

def \_\_init\_\_(self, name, id, salary, address):

self.name = name

self.id = id

self.salary = salary

self.address = address

#Display state of object

def display(self):

print(self.name,"\t",self.id,"\t",self.salary,"\t",self.address)

# with keyword - not required to close

with open("emp.dat","wb") as f:

emp1 = Employee("Suzan", 1672, 25000, "California")

pickle.dump(emp1, f)

print("Pickling of Employee object completed.")

#unplicking

with open("emp.dat","rb") as f:

obj = pickle.load(f)

print("Employee Information afte unplicking..")

obj.display()

2. How Memory Management is achieved in Python? Explain with the help of example.

Memory Management in Python is achieved by using Python private heap space. All Python objects and data structures are stored in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead. The allocation of heap space for Python objects is done by Python’s memory manager. The core API gives access to some tools for the programmer to code. Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

3. Write a Program in Python to explain Multithreading in Python.

#Multithreading library is lightweight, shares memory, resposible for responsive UI

#for I/O bound applications

import time

import threading

def calc\_square(nums):

print("Calculate square numbers")

for n in nums:

time.sleep(0.2)

print("square : ", n\*n)

def calc\_cube(nums):

print("Calculate cube of numbers")

for n in nums:

time.sleep(0.2)

print("cube : ", n\*n\*n)

lst = [2,4,6,9]

t = time.time()

thread1 = threading.Thread(target=calc\_square,args=(lst,))

thread2 = threading.Thread(target=calc\_cube,args=(lst,))

#Execution of both function in parallel

thread1.start()

thread2.start()

#Join thread back to the parent process

thread1.join()

thread2.join()

calc\_square(lst)

calc\_cube(lst)

4. What are the Collection in Python. Explain following terms with Example.

·  [namedtuple()](https://docs.python.org/3/library/collections.html#collections.namedtuple)

·  [Counter](https://docs.python.org/3/library/collections.html#collections.Counter)

·  [OrderedDict](https://docs.python.org/3/library/collections.html#collections.OrderedDict)

· C[hainMap](https://stackabuse.com/introduction-to-pythons-collections-module/#thechainmap)

·  [deque](https://stackabuse.com/introduction-to-pythons-collections-module/#thedeque)

NOTE: Make sure to write at least one program on each and comment what each line of the code is doing.

Collections in Python are containers that are used to store object. They provide a way to access the contained objects and iterate over them. Example of built-in object include Tuple, list and dictionary. Python Collections modules are some of the modules that have been developed to provide additional data structures to store collections of data.

[namedtuple()](https://docs.python.org/3/library/collections.html#collections.namedtuple) ~ It returns a tuple with name for each position in the tuple.

#import namedtuple from the collections module

from collections import namedtuple

#create a namedtuple

Student = namedtuple('Student', 'fname, lname, age')

s1 = Student('John', 'Clarke', '13')

print(s1.fname)

[Counter](https://docs.python.org/3/library/collections.html#collections.Counter)~ A Counter is a container that keeps track of how many times equivalent values are added. It is a subclass of dictionary object.

#import Counter class from the collections module

from collections import Counter

#create Counter object

list = [1,2,3,4,1,2,6,7,3,8,1]

cnt = Counter(list)

print(cnt[1])

[OrderedDict](https://docs.python.org/3/library/collections.html#collections.OrderedDict) ~ OrderedDict is a dictionary where keys maintain the order in which they are inserted, which means if you change the value of a key later, it will not change the position of the key.

#import OrderedDict from the collections module

from collections import OrderDict

#create OrderedDict object with OrderedDict constructor without any arguments

od = OrderedDict()

od['a'] = 1

od['b'] = 2

od['c'] = 3

print(od)

ChainMap ~ ChainMap is used to combine several dictionaries or mappings, and returns a list of dictionaries.

#import a chainmap from the collections module

from collections import ChainMap

#Use of ChainMap() constructor, and pass the dictionaries

dict1 = { 'a' : 1, 'b' : 2 }

dict2 = { 'c' : 3, 'b' : 4 }

chain\_map = ChainMap(dict1, dict2)

print(chain\_map.maps)

deque ~ The deque is a list optimized for inserting and removing items.

#import deque from collections module

from collections import deque

#create deque with deque() constructor, and pass a list as an argument

list = ["a","b","c"]

deq = deque(list)

print(deq)

5. Learn about PEP8 - How to write beautiful Python Code?

6. Difference between the pip, virtualenv, conda?

Python has three ways of creating virtual environment at the moment- pip, virtualenv and conda.

Pip is the Python Packaging Authority’s recommended tool for installing packages from the Python Package Index, PyPI, additionally installing python software packaged as wheel. Conda is a cross platform package and environment manager that installs and manages conda packages from the Anaconda repository as well as from the Anaconda Cloud. Conda packages are binaries. Virtualenv is a tool to create isolated Python environments.

The difference between conda and pip is that pip install python packages whereas conda installs packages which may contain software written in any language.

**WEEKEND ACTIVITY- GITHUB TASK**

[(https://drive.google.com/file/d/1E8MY1ML\_SsYDmN\_MFwAgbApur\_1OqqLQ/view?usp=sharing)](https://drive.google.com/file/d/1E8MY1ML_SsYDmN_MFwAgbApur_1OqqLQ/view?usp=sharing))

1. What is Version Control System and their types of tools available mentioned any top 5?.

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Top 5 available version control system are

· Git

· Microsoft Team Foundation Server

· AWS CodeCommit

· HelixCore

· Bitbucket

2. What is the difference between Git and GitHub?

Git is a distributed peer to peer version control system, a tool to manage source code history while GitHub s a web-based hosting service for git repository which provides access control and several collaboration features such as task management and bug tracking for all features.

3. Answer the following question:

· PR is Version Control Systems

Pull Requests(PR) in a version control system is a recent changes made that is pushed to GitHub repository. Once PR is sent, collaborators in a project can review the set of changes, make potential modification, and even push follow-up commits if necessary.

· Fork VS Commit

Git fork is a copy of a repository that allows to freely experiment with changes without affecting the original project. While Git commit adds the latest changes to the local repository.

· Init VS Clone

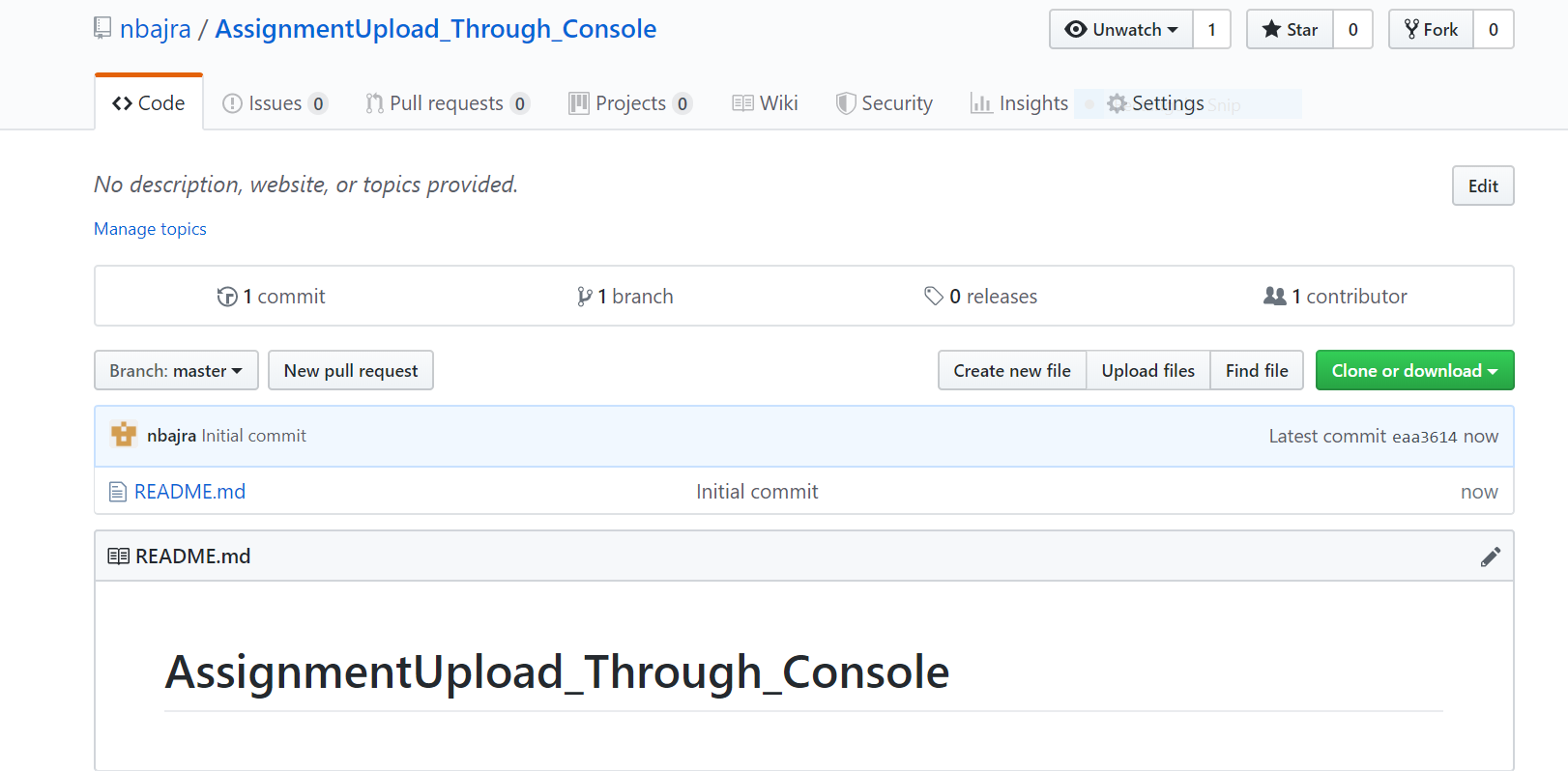
Git Init is used to initialize and create empty git repository in a local machine while git clone is used to make a copy of an existing git repository to system.

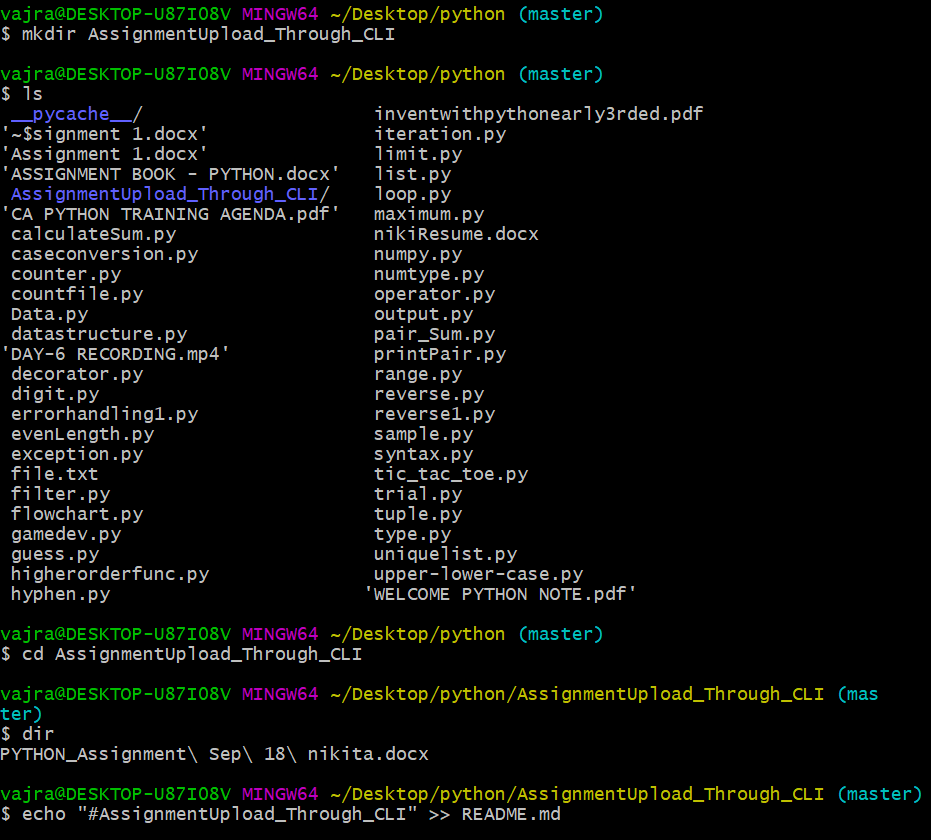
· Branching and Merging

Branching is GitHub is a way to create a separate branch to develop a feature(or work on a bug) without disturbing the master branch. We can make new branch by using command “git branch new\_feature”

Merging is a way to put a forked history back together again. The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

4. Create a Git Repositories using CLI and Console make sure to give them a name AssingmentUpload\_Through\_CLI and AssignmentUpload\_Through\_Console respectively.





5. Make sure to upload all your assignment into both of your repositories again using CLI and Console.

HINT: Use following Commands for CLI

· git Init

· git add

· git commit -m “Message”

· git status

· git push

· git config --global user.name "FIRST\_NAME LAST\_NAME"

· git config --global user.email "MY\_NAME@example.com"

6. Make sure to create three new branches and follow the instructions given below:

· Main Branch - **Master**

· First Branch - **Dev** ---> Copy of **Master**

· Second Branch - **Test** ---> Copy of **Dev**

· First Branch - **Prod** ---> Copy of **Master**

· Once you are done creating temp branches see how to delete them but make sure delete them only when the changes have been pushed to the master branch or any main branch.