PseudoCode

Variables

customer\_name = “ “

Balance= 0,0

Count=0.0

#Creating class BankAccount

class BankAccount

customer\_name

balance

count

\_\_init\_\_(customer\_name)

#inislize customer\_name

#Generating account\_num using generateAccountNumber() method

account\_num = generateAccountNumber()

balance =0

#Calling display method

display()

#increment the count

count+1

/#This method creates random account Number and return the value

generateAccountNumber()

return 1+random.randint(0,1000)\*1000+random.randint(0,1000)

addAmount(amount)

#add amount to balance

balance += amount

withdrawAmount(amount)

if(check for balance-amount should not be negative)

subtract amount from balance

else

print(Not sufficient balance)

display()

print(customer\_name,account\_num,balance)

#Create Empty Bank account Collection

BankAccountCollection = []

Variable to count of entered account

count = 1

#method to create new Back Account

opennewAccount()

name = str(input("..."))

#Creating new BankAccount object with name

obj = BankAccounts(name)

Adding BankAccount to Collection

increment count Variable

#method to close account

closeAccount()

account\_number = int(input(....))

try:

for i in range(count):

if BankAccountCollection[i].getAccountNumber() is equal to account\_number

display bank account

BankAccountCollection[i].display()

#remove back account Collection by calling remove method

BankAccountCollection.remove(BankAccountCollection[i])

decrease value of count

break the loop

except ValueError as ve:

print(Enter valid account number)

except IndexError as e

print(Account does not exist)

displayAccountDetails()

try:

account\_number = int(input(...))

for i in range(count)

if BankAccountCollection[i].getAccountNumber() is equal to account\_number

display bank account

BankAccountCollection[i].display()

break the loop

except ValueError as ve:

print(Enter valid account number)

except IndexError as e

print(Account does not exist)

depositeAmount()

try

account\_number = int(input(...))

for i in range(count)

if BankAccountCollection[i].getAccountNumber() is equal to account\_number

display bank account

BankAccountCollection[i].display()

amount = int(input(....))

check amount is greatter than zero and add amount to account

break the loop

except ValueError as ve:

print(Enter valid account number)

except IndexError as e

print(Account does not exist)

withdrawAmount()

try

account\_number = int(input(...))

for i in range(count)

if BankAccountCollection[i].getAccountNumber() is equal to account\_number

display bank account

BankAccountCollection[i].display()

amount = int(input(....))

check amount is greatter than zero and withdraw amount to account

break the loop

except ValueError as ve:

print(Enter valid account number)

except IndexError as e

print(Account does not exist)