PROGRAM 4: EXCEPTION HANDLING IN PYTHON

Requirements:

Little Johnny the Gen-Z brat has enrolled himself in WhiteHat Junior,online coding platform.His father wants to test his coding skills. Here is the task related to exception handling given to Johnny by his father as a rhyme. Help e out to accomplish the task successfully.

TASK:

👦: Johnny Johnny

👶: Yo Dada

👦: Trying Code? 💻

👶: Yo Dada

👦: Done with Exceptions?

👶: No Dada

👦: Do it now...

👶: Ya,Ya,Ya

In [2]:
```
data={}
```

In [3]:
```
#importing libraries
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

In [4]:
```
class Number_invalid(Exception):
    def __init__(self):
        print("Your contact number should start with 6,7,8 or 9")
```

In [5]:
```
#Account phone number
def accountholder_phone():
    k=list(('6','7','8','9'))
    t=True
    phone=None
    while(t):
        try:
            print("_____")
            phone=input('Phone number : ')
            if phone[0] not in k:
                raise Number_invalid()
            assert phone.isnumeric(),print("Invalid phone number")
            assert len(phone)==10,print("Invalid length")
        except Number_invalid:
            print("Invalid phone, Re-enter your phone number")
            t=True
        except AssertionError:
            print('Invalid phone, Re-enter your phone number')
            t=True
        except (ValueError,NameError):
            print("Invalid phone, Re-enter your phone number")
            t=True
        except KeyboardInterrupt:
            print('*mandatory, please fill your contact info')
            t=True
        else:
            t=False
            return phone
```

In [6]:

```
#Account aadhar number
def accountholder_aadhar():
    t=True
    adhar=None
    while(t):
        try:
            print("_____")
            adhar=input('Aadhar number : ')
            assert adhar.isnumeric(),print("Invalid type")
            assert len(adhar)==12,print("Invalid Aadhar number")
        except AssertionError:
            print('Invalid Aadhar, please enter your Aadhar details')
            t=True
        except (ValueError,NameError):
            print("Invalid Aadhar, please enter your Aadhar details")
            t=True
        except KeyboardInterrupt:
            print ('*mandatory, please fill your Aadhar detail')
            t=True
        else:
            t=False
            return adhar
```

In [7]:

```
#Account holder name
def accountholder_name():
    t=True
    first=None
    last=None
    while(t):
        try:
            print("_____")
            first=input('First name : ')
            first=first.upper()
            assert first.isalpha(),"Invalid character entry"
            last=input('Last name : ')
            last=last.upper()
            assert last.isalpha(),"Invalid character entry"
        except AssertionError:
            print('Invalid name, please enter your name')
            t=True
        except KeyboardInterrupt:
            print ('*mandatory, please fill your name')
            t=True
        else:
            t=False
            return first,last
```

In [8]:
```python
#Account holder age
def accountholder_age():
    t=True
    age=None
    while(t):
        try:
            print("_____")
            age=int(input('Age : '))
            assert age>0 and age<115,print("Oops!, please give valid age")
        except (ValueError,AssertionError):
            print("Oops!, please provide your valid age")
            t=True
        except KeyboardInterrupt:
            print ('*mandatory, Please fill your age details')
            t=True
        else:
            t=False
            return age
```

In [9]:
```python
# Exception handling for gender
class GenderInvalid(Exception):
    def __init__(self):
        print('Please provide valid gender [female/male/others]')
```

In [10]:
```python
# Account holder gender
def accountholder_gender():
    g=list(('female','male','others'))
    Gender=None
    c1=True
    while(c1):
        try:
            print("_____")
            Gender=str(input("Gender: "))
            Gender=Gender.lower()
            if Gender not in g:
                raise GenderInvalid()
        except GenderInvalid as e2:
            print('Something went wrong!, Invalid Gender')
            c1=True
        else:
            c1=False
            return Gender
```

In [11]:
```python
# Exception handling for account type
class AccountInvalid(Exception):
    def __init__(self):
        print('Please check your Account Type')
```

In [12]:

```
# Account type
def account_type():
    acc=list(('current','savings'))
    acc_type=None
    c1=True
    while(c1):
        try:
            print("_____")
            print("Account type are : [Current/Savings]")
            acc_type=str(input("Account type: "))
            acc_type=acc_type.lower()
            if acc_type not in acc:
                raise AccountInvalid()
        except AccountInvalid as e2:
            print('Something went wrong!, provide valid Account type [Current/Savings]')
            c1=True
        except KeyboardInterrupt:
            print ('*mandatory, Provide Account Type')
            c1=True
        else:
            c1=False
            return acc_type
```

In [13]:
```python
# Account's Initial deposit
def initial_deposit(acc_type):
    t=True
    bal=None
    typ=acc_type
    typ=typ.lower()
    while(t):
        try:
            print("_____")
            if (typ=='current'):
                print("Please deposit mininum of ₹1000 for CURRENT ACCOUNT\n")
                bal=float(input('Initial deposit ₹: '))
                assert int(bal)>=1000,print('Invalid : Initial deposit not vaild ')
            elif (typ=='savings'):
                print("Please deposit mininum of ₹500 for SAVINGS ACCOUNT\n")
                bal=float(input('Initial deposit ₹:'))
                assert int(bal)>500,print('Invalid : Initial deposit not vaild ')
        except Exception:
            t=True
            print('Enter the initial deposit properly in ₹')
        except KeyboardInterrupt:
            print ('Minimum deposit is mandatory')
            t=True
        else:
            t=False
            return float(bal)
```

In [14]:
```
# Account's Money deposit
def money(ip):
    t=True
    bal=None
    while(t):
        try:
            print("_____")
            print("Enter The money {} >=0 in ₹\n".format(ip))
            bal=float(input('Amount ₹: '))
            assert int(bal)>=0,print('Invalid amount, please try again!')
        except Exception:
            t=True
            print('Enter the money {} properly in ₹'.format(ip))
        except KeyboardInterrupt:
            print ('*mandatory, please fill appropriate detail')
            t=True
        else:
            t=False
            return bal
```

In [15]:
```
import random
def account_number():
    ac='JB'
    for len in range(1,13):
        ac=ac+str(random.randint(0,9))
    return (ac)
```

In [16]:
```
def account_password():
    ac=''
    for len in range(1,9):
        ac=ac+str(random.randint(0,9))
    return int(ac)
```

In [17]:

```
# Create account
def create_account(data):
    name=accountholder_name()
    age=accountholder_age()
    gender=accountholder_gender()
    phone=accountholder_phone()
    adhar=accountholder_aadhar()
    acc_type=account_type()
    pword=None
    pword=account_password()
    accno=None
    t=True
    while(t):
        acno=account_number()
        if acno in data.keys():
            t=True
        else:
            t=False

    i_d=initial_deposit(acc_type)
    print(i_d)
    data[acno]=[name,age,gender,acc_type,i_d,pword,phone,adhar]
    print()
    print("\t\t"+"*"+'-'*50+"*")
    print("\t\t"+"|{:^50s}|".format("Welcome {}".format(' '.join(name))))
    print("\t\t"+"*"+'-'*50+"*")
    print("\t\t"+"|{:^50s}|".format("Account Number : {}".format(acno)))
    print("\t\t"+"|{:^50s}|".format("Account Type   : {}".format(acc_type)))
    print("\t\t"+"|{:^50s}|".format("Account Holder : {}".format(' '.join(name))))
    print("\t\t"+"|{:^50s}|".format("Password       : {}".format(pword)))
    print("\t\t"+"*"+'-'*50+"*")
    print("\t\t\t**********\033[1m"+" Account Created"+"\033[0m**********")
    print()
    print("_____________________________________________")
    return data
```

In [18]:
```
def acc_no_entry():
    t=True
    while(t):
        try:
            print("\033[1m"+"LOG IN"+"\033[0m")
            print("_____________________________________________")
            print()
            print("Account number:")
            n=input()
            num=n[2:]
            num=int(num)
            if not (num<0 or len(str(n))==14):
                raise ValueError()
        except ValueError:
            print('Account Number not found or Invalid, provide valid Account details :(')
            t=True
        except KeyboardInterrupt:
            print ('*mandatory, please fill your ACCOUNT NUMBER')
            t=True
        else:
            t=False
            return str(n)
```

In [19]:
```
import getpass
def passwrd():
    try:
        print("Password:")
        p=getpass.getpass()
    except Exception as error:
        print('ERROR', error)
    else:
        return int(p)
```

In [20]:
```
# Login account
def login_account(data):
    t=True
    user_id=None
    while(t):
        user_id=acc_no_entry()
        password=passwrd()
        type(password)
        if (user_id in data.keys() and (data[user_id][5]==password)):
            t=False
            return user_id
        else:
            t=True
            print("Invalid User ID & Password ")
```

In [21]:
```
# Deposit
def deposit(data,user_id):
    deposit=money('to deposit')
    data[user_id][4]+=deposit
    print("\n\033[1m"+"Money dopsited sucessfully.\nFinal balance :"+"\033[0m",data[user_id][4])
```

In [53]:
```
# Exception handling for balance error
class BalanceInvalid(Exception):
    def __init__(self):
        print("\033[1m"+"Insufficient account balance"+"\033[0m")
```

In [54]:
```
# Exception handling for denomination error
class DenominationInvalid(Exception):
    def __init__(self):
        print("\033[1m"+"Denomination should be multiple of 100"+"\033[0m")
```

In [59]:

```
# Withdrawal
import datetime
def withdrawal(data,user_id):
    acc_no=user_id
    acc_type=data[user_id][3]
    mon=None
    m=None

    if (acc_type=='savings'):
        t=True
        while(t):
            try:
                mon=money('to withdraw')
                bal=data[acc_no][4]
                m=bal-mon

                if m<=500:
                    raise BalanceInvalid()
                if m%100!=0:
                    raise DenominationInvalid()
                else:
                    data[acc_no][4]= m
            except (BalanceInvalid,DenominationInvalid):
                print('Oops!')
                t=True
            else:
                t=False


    elif (acc_type=='current'):
        t=True
        while(t):
            try:
                mon=money('to withdraw')
                bal=data[acc_no][4]
                m=bal-mon

                if m<=1000:
                    raise BalanceInvalid()
                if m%100!=0:
                    raise DenominationInvalid()
```

```
                else:
                    data[acc_no][4]= m
            except (BalanceInvalid,DenominationInvalid):
                print('Oops!')
                t=True
            else:
                t=False

    else:
        print("Invalid")



    dates=datetime.datetime.now()
    date=dates.strftime("%x")
    time=dates.strftime("%X")
    #print_details(data,acc_no,mon,date,time)

    print("\t\t"+"*"+'-'*40+"*")
    print("\t\t"+"|{:^40s}|".format("Transaction Date : {}".format(date)))
    print("\t\t"+"|{:^40s}|".format("Transaction Time : {}".format(time)))
    print("\t\t"+"|{:^40s}|".format("Amount taken ₹   : {}".format(mon)))
    print("\t\t"+"|{:^40s}|".format("Balance amount ₹ : {}".format(data[acc_no][4])))
    print("\t\t"+"*"+'-'*40+"*")
```

In [24]:

```
# Balance Enquiry
def balance_enquiry(data,user_id):
    acc_no=user_id
    dates=datetime.datetime.now()
    date=dates.strftime("%x")
    time=dates.strftime("%X")
    #print_details(data,acc_no,mon,date,time)
    print("\t\t"+"*"+'-'*40+"*")
    print("\t\t"+"|{:^40s}|".format("Account Number   : {}".format(acc_no)))
    print("\t\t"+"|{:^40s}|".format("Transaction Date : {}".format(date)))
    print("\t\t"+"|{:^40s}|".format("Transaction Time : {}".format(time)))
    print("\t\t"+"|{:^40s}|".format("Balance amount ₹ : {}".format(data[acc_no][4])))
    print("\t\t"+"*"+'-'*40+"*")
```

In [25]:
```
# Account detials
def Account_details(data,user_id):
    acc_no=user_id
    dates=datetime.datetime.now()
    date=dates.strftime("%x")
    time=dates.strftime("%X")

    print("\t\t"+"*"+'-'*50+"*")
    print("\t\t"+"|{:^58s}|".format("\033[1m"+"ACCOUNT DETAILS"+"\033[0m"))
    print("\t\t"+"*"+'-'*50+"*")
    print("\t\t"+"|{:^50s}|".format("Account Number   : {}".format(acc_no)))
    print("\t\t"+"|{:^50s}|".format("Account Type     : {}".format(data[acc_no][3])))
    print("\t\t"+"|{:^50s}|".format('Account Holder   : {}'.format(' '.join(data[acc_no][0]))))
    print("\t\t"+"|{:^50s}|".format('Aadhar Number    : {}'.format(data[acc_no][7])))
    print("\t\t"+"|{:^50s}|".format('Customer Age     : {}'.format(data[acc_no][1])))
    print("\t\t"+"|{:^50s}|".format('Customer Gender  : {}'.format(data[acc_no][2])))
    print("\t\t"+"|{:^50s}|".format('Customer Phone   : {}'.format(data[acc_no][6])))
    print("\t\t"+"|{:^50s}|".format('Enquiry Date     : {}'.format(date)))
    print("\t\t"+"|{:^50s}|".format('Enquiry Time     : {}'.format(time)))
    print("\t\t"+"|{:^50s}|".format('Balance amount ₹ : {}'.format(data[acc_no][4])))
    print("\t\t"+"*"+'-'*50+"*")
```

In [26]:
```
# Account update
def update_account(data,user_id):
    acc_no=user_id
    print("1.Phone Number")
    print("2.Aadhar Number")
    ch=input("Enter your option:")
    if ch=='1':
        data[acc_no][6]=accountholder_phone()
    elif ch=='2':
        data[acc_no][7]=accountholder_aadhar()
    else:
        print("Not a valid option")
```

In [27]:

```
# menu code
def menu_ch(data,user_id):
    data=data
    user_id=user_id
    pictures = ["C:/Users/Admin/Desktop/pictures/Bank/1.jpg","C:/Users/Admin/Desktop/pictures/Bank/4.PNG",
                    "C:/Users/Admin/Desktop/pictures/Bank/00.PNG","C:/Users/Admin/Desktop/pictures/Bank/5.jpg"]
    op='yes'
    while(op=='yes'):
        fig=plt.figure(figsize=(10,12))
        pic = pictures[2]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        print()
        ch=input("Enter your option: ")
        if ch == '1':
            deposit(data,user_id)
        elif ch =='2':
            withdrawal(data,user_id)
        elif ch == '3':
            balance_enquiry(data,user_id)
        elif ch == '4':
            Account_details(data,user_id)
        elif ch == '5':
            update_account(data,user_id)
        elif ch == '6':
            fig=plt.figure(figsize=(15,15))
            pic = pictures[3]
            img=mpimg.imread(pic)
            plt.imshow(img)
            plt.axis('off')
            plt.show("\n")
            break
        else :
            print("Please select your option from our services")

        op = input("Do you continue our services[yes/no] : ")
```

In [32]:

```
# main code
def Johnny_Bank(data):
    print()
    pictures = ["C:/Users/Admin/Desktop/pictures/Bank/1.jpg","C:/Users/Admin/Desktop/pictures/Bank/000.PNG",
                "C:/Users/Admin/Desktop/pictures/Bank/00.PNG","C:/Users/Admin/Desktop/pictures/Bank/5.jpg"]
    op='yes'
    while(op=='yes'):
        fig=plt.figure(figsize=(15,15))
        pic = pictures[0]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        print("\n\n")

        print("Johnny Bank is the consumer division of financial services multinational Johnny group. Johnny Bank was fo
        print("\n\tCustomer service   : 1860 210 2484")
        print("\tCEO                : Mr.Jimmy [Johnny's Dad] ")
        print("\tHeadquarters       : New York, New York, United States")
        print("\tParent organization: Johnny group")
        print("\tFounder            : Mr.Johnny")

        fig=plt.figure(figsize=(8,10))
        pic = pictures[1]
        img=mpimg.imread(pic)
        plt.imshow(img)
        plt.axis('off')
        plt.show("\n")
        print()
        c=input("Enter your option number: ")
        if c == '1':
            user_id=login_account(data)

        elif c == '2':
            data=create_account(data)
            user_id=login_account(data)
        else:
            print("Please select your option from our services")
        menu_ch(data,user_id)
        op = input("Do you continue [yes/no] : ")
```

In [63]:
```
Johnny_Bank(data)
```

Johnny Bank is the consumer division of financial services multinational Johnny group. Johnny Bank was founded in 2020.

```
Customer service   : 1860 210 2484
CEO                : Mr.Jimmy [Johnny's Dad]
Headquarters       : New York, New York, United States
Parent organization: Johnny group
Founder            : Mr.Johnny
```

1. Login	2. Create Account

```
Enter your option number: 1
```
LOG IN

Account number:
JB322380488289
Password:
.....



1. Deposit	2. Withdrawal	3. Balance enquiry



4. Account details	5. Update Account	6. Log out

Enter your option: 2

Enter The money to withdraw >=0 in ₹

Amount ₹: 20000
Insufficient account balance
Oops!

Enter The money to withdraw >=0 in ₹

Amount ₹: 2999

Insufficient account balance
Oops!

_____
Enter The money to withdraw >=0 in ₹

Amount ₹: 199
Denomination should be multiple of 100
Oops!

_____
Enter The money to withdraw >=0 in ₹

Amount ₹: 200

```
                *---------------------------------------*
                |      Transaction Date : 09/29/20      |
                |      Transaction Time : 12:47:52      |
                |        Amount taken ₹   : 200.0       |
                |       Balance amount ₹ : 1100.0       |
                *---------------------------------------*
```

Do you continue our services[yes/no] : yes



1. Deposit	2. Withdrawal	3. Balance enquiry



4. Account details	5. Update Account	6. Log out

```
Enter your option: 5
1.Phone Number
2.Aadhar Number
Enter your option:1
_____________________________________
Phone number : 7876789898
Do you continue our services[yes/no] : yes
```



1. Deposit	2. Withdrawal	3. Balance enquiry



```
Enter your option: 1
_____________________________________________
Enter The money to deposit >=0 in ₹

Amount ₹: 2990

Money dopsited sucessfully.
Final balance : 4090.0
Do you continue our services[yes/no] : yes
```



```
Enter your option: 4
                *-------------------------------------------------*
                |                 ACCOUNT DETAILS                 |
                *-------------------------------------------------*
                |        Account Number   : JB322380488289        |
                |           Account Type     : current            |
                |          Account Holder   : SOUNDARYA G         |
                |          Aadhar Number    : 123467898789        |
                |              Customer Age     : 21              |
                |           Customer Gender  : female             |
                |          Customer Phone   : 7876789898          |
                |           Enquiry Date     : 09/29/20          |
                |           Enquiry Time     : 12:48:54          |
                |            Balance amount ₹ : 4090.0            |
                *-------------------------------------------------*
Do you continue our services[yes/no] : yes
```



```
Enter your option: 3
                *----------------------------------------*
                |   Account Number   : JB322380488289    |
                |      Transaction Date : 09/29/20       |
                |      Transaction Time : 12:49:13       |
                |       Balance amount ₹ : 4090.0        |
                *----------------------------------------*
Do you continue our services[yes/no] : yes
```



1. Deposit	2. Withdrawal	3. Balance enquiry



4. Account details	5. Update Account	6. Log out

Enter your option: 6

Do you continue [yes/no] : no