CAMBRIDGE 0984 / 0478 IGCSE COMPUTER SCIENCE
PAPER 2 REVISION EXTENDED PROGRAMMING QUESTION

DIANA EDWIN
TUTORING

BANK ACCOUNT - PYTHON MODEL ANSWER WITH PROCEDURES

```
AccID = int(input("Please enter your account number: "))
if CheckDetails(AccID) == True:     # This calls the CheckDetails() procedure
    Choice = 0
    Exit = False
    while Exit != True:
        print("Menu Options")
        print("1. Display balance")
        print("2. Withdraw money")
        print("3. Deposit money")
        print("4. Exit")
        Choice = int(input("Please choose 1, 2, 3 or 4: "))
        if Choice == 1:
            Balance(AccID)    # This calls the Balance() procedure
        elif Choice == 2:
            Withdraw(AccID)   # This calls the Withdraw() procedure
        elif Choice == 3:
            Deposit(AccID)    # This calls the Deposit() procedure
        elif Choice == 4:
            Exit = True
        else:
            print("Invalid choice")

def CheckDetails (ID):        # This procedure checks account number, name and password
    Valid = False
    if ID < 0 or ID >= Size:
        print("Invalid Account Number")
    else:
        Name = input("Please enter name: ")
        Password = input("Please enter password: ")
        if Account[ID][0] != Name:
            print("Invalid name.")
        elif Password != Account[ID][1]:
            print("Invalid password.")
        else:
            Valid = True
    return Valid

def Balance(Acc):             # This procedure outputs the balance of the user's account
    print("Your balance is", AccDetails[Acc][0])

def Withdraw(Acc):            # This procedure checks that a withdrawal can be made
    Amount = -999.99 # initialising value so loop happens at least once
    while (Amount > AccDetails[Acc][2] or Amount > AccDetails[Acc][1] + AccDetails[Acc][0]
                or Amount < 0):
        Amount = float(input("Please enter amount to withdraw: "))
        if Amount > AccDetails[Acc][2]:
            print("Amount greater than withdrawal limit.")
        if Amount > AccDetails[Acc][1] + AccDetails[Acc][0]:
            print("Amount greater than cash available.")
        if Amount <= AccDetails[Acc][2] and Amount < AccDetails[Acc][1] + AccDetails[Acc][0]:
            AccDetails[Acc][0] -= Amount #decrement by the value of Amount

def Deposit(Acc):             # This procedure allows the account holder to deposit money
    Amount = -999.99
    while Amount <= 0:
        Amount = float(input("Please enter a positive amount to deposit: "))
    AccDetails[Acc][0] += Amount
```

CAMBRIDGE 0984 / 0478 IGCSE COMPUTER SCIENCE
PAPER 2 REVISION EXTENDED PROGRAMMING QUESTION

DIANA EDWIN
TUTORING

BANK ACCOUNT - PSEUDOCODE MODEL ANSWER WITH PROCEDURES

```
OUTPUT "Please enter your account number: "
INPUT AccountNumber
CheckDetails(AccountNumber)                    # This calls the CheckDetails() procedure
IF Valid = TRUE THEN
    REPEAT
        OUTPUT "Menu Options - Please choose 1, 2, 3 or 4:"
        OUTPUT "1. Display balance"
        OUTPUT "2. Withdraw money"
        OUTPUT "3. Deposit money"
        OUTPUT "4. Exit"
        INPUT Choice
        CASE OF Choice
            1 : Balance(AccountNumber)          # This calls the Balance() procedure
            2 : Withdraw(AccountNumber)       # This calls the Withdraw() procedure
            3 : Deposit(AccountNumber)          # This calls the Deposit() procedure
            4 : Exit <- TRUE
            OTHERWISE OUTPUT "Invalid choice"
        ENDCASE
    UNTIL Exit = TRUE
ELSE
    OUTPUT "Invalid account number."
ENDIF

PROCEDURE CheckDetails (AccID: INTEGER)
    DECLARE Name, Password : STRING // Local variables
    Valid <- FALSE
    IF AccID < 0 OR AccID > Size THEN
        OUTPUT "Invalid Account Number"
    ELSE
        OUTPUT "Please enter name: "
        INPUT Name
        OUTPUT "Please enter password: "
        INPUT Password
        IF Name <> Account[AccID, 1] OR Password <> Account[AccID, 2] THEN
            OUTPUT "Invalid name or password."
        ELSE
            Valid <- TRUE
        ENDIF
    ENDIF
ENDPROCEDURE

PROCEDURE Balance(AccID: INTEGER)
    OUTPUT "Your balance is", AccDetails[AccID, 1]
ENDPROCEDURE

PROCEDURE WithDraw(AccID : INTEGER)
    DECLARE Amount : REAL // local variable
    REPEAT
        OUTPUT "Please enter amount to withdraw: "
        INPUT Amount
        IF Amount > AccDetails[AccID,3] THEN
            OUTPUT "Amount greater than withdrawal limit."
        ENDIF
        IF Amount > AccDetails[AccID,2] + AccDetails[AccID,1] THEN
            OUTPUT "Amount greater than cash available."
        ENDIF
        IF Amount <= AccDetails[AccID,3] AND Amount < AccDetails[AccID,2] + AccDetails[AccID,1] THEN
            AccDetails[AccID,1] <- AccDetails[AccID,1] - Amount
        ENDIF
    UNTIL Amount <= AccDetails[AccID,3] AND Amount <= AccDetails[AccID,2] + AccDetails[AccID,1]
            AND Amount > 0
ENDPROCEDURE

PROCEDURE Deposit(AccID : INTEGER)
    DECLARE Amount : REAL // local variable
    REPEAT
        OUTPUT "Please enter a positive amount to deposit: "
        INPUT Amount
    UNTIL Amount > 0
    AccDetails[AccID,1] <- AccDetails[AccID,1] + Amount
ENDPROCEDURE
```