INFORMATICS INSTITUTE OF TECHNOLOGY

Foundation Certificate in Higher Education

Module:	DOC333
Module name:	Introduction to programming principals
Module Leader:	Mr. Sudharshan Walihinda
Assignment Number:	1
Type of Assignment:	Individual
Submission Date:	19th July 2024
Student Name:	W.D. Nisitha Nimsara
Student ID:	20240281

2. Acknowledgement

I am profoundly grateful to the IIT management for organizing the foundation program and creating a helpful environment for us to improve our academic and general skills under the direction of our reputable lecturers.

First and foremost, I would like to express my gratitude to Mr. Sudharshan Walihinda, Module leader, for his invaluable guidance and unwavering support during this project. I would also like to extend my appreciation to Mr. Thamara waidyarathna and Ms. Imasha Umayangi, lecturers of the module “DOC333: Introduction to Programming Principals”, for their constructive feedback and encouragement throughout this project. Their guidance and input have been crucial in shaping the direction and improving the quality of our work. Finally, I want to express my special thanks to my batchmates and friends, for providing technical advice to me and sharing the knowledge they have. Which have been fundamental in helping me to reach this milestone.

3. Table of contents

Table of Contents

4. List of Figures

5. List of tables

6. Introduction

Design an information system for ABC bank in Sri Lanka to register their customer details and manage those details for their savings account holders. This system support to, open new account and display customer or all customers' details with their account number, name and the bank balance when requested .And it supports to update customer details (except account number and bank balance) already registered in the system. Also, this system can manage transactions including deposit and withdraw from given accounts.

7. Explain the algorithm

7.1. Lists

As the first step, create **lists** for each category and set them **empty** . So that lists will store data when user input them and that lists structure will be useful to returns values when requested.

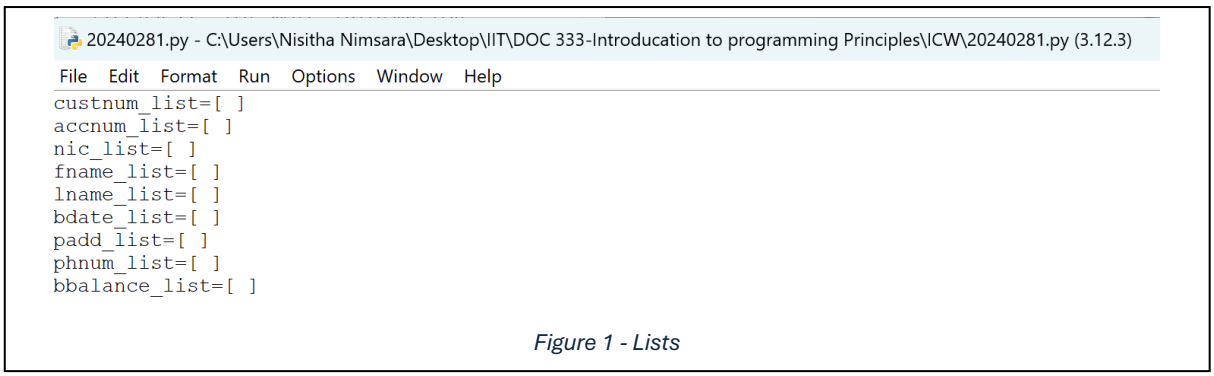


Figure 1 - Lists

7.2. Add a new customer function

To create a function for “Add a new customer” option, **def** keyword uses and rename the function as “customerinput” then use **print** function to add relevant details to interface. After that set variables to get different data types of input from user and then turn all data type into **string** for get character length(**len**) to assign limit to each input using **if** condition. If user enter wrong character count or wrong data type to irrelevant location that will display hints for correct inputs and gave option to try again or go back to main Manu. After that user input all details correctly, then “Do you want to save the account (Yes/No)?” option displays.

```
#-------------------------------------------------------------------------------------------------------------
def customerinput ():               # -1- customer detail input function
    print(" ")
    print("                        ""ABC Bank")
    print("                    ""Add a new customer")
    print(" ")
    try:
        bnk_acc_num = int(input("Bank Account Number       - "))
        test = str(bnk_acc_num)
        if (len(test) == 10):
            nic = str(input("NIC                        - "))
            test = str(nic)
            if (len(test) == 10):
                f_name = input("First Name                - ")
                test = str(f_name)
                if (len(test) < 11):
                    l_name = input("Last Name                 - ")
                    test = str(l_name)
                    if (len(test) < 16):
                        birth_date = str(input("Birth Date                - "))
                        test = str(birth_date)
                        if (len(test) < 18):
                            pmnt_add = input("Permanent Address         - ")
                            test = str(pmnt_add)
                            if (len(test) < 16):
                                ph_num = str(input("Phone Number              - "))
                                test = str(ph_num)
                                if (len(test) == 10):
                                    bank_balance = float(0)
                                else:
                                    print("_____________________________________________________________________")
                                    print("Phone Number must be 10 Digit.")
                                    trycustomerin()
                            else:
                                print("_____________________________________________________________________")
                                print("Permanent Address maximum of 15 characters.")
                                trycustomerin()
                        else:
                            trycustomerin()
                    else:
                        print("_____________________________________________________________________")
                        print("Lasa Name maximum of 15 characters.")
                        trycustomerin()
                else:
                    print("_____________________________________________________________________")
                    print("First Name maximum of 10 characters.")
                    trycustomerin()
            else:
                print("_____________________________________________________________________")
                print("NIC must be 10 digits.")
                trycustomerin()
        else:
            print("_____________________________________________________________________")
            print("Account Number must be 10 digits.")
            trycustomerin()

        print(" ")
        answer = input("Do you want to save the account (Yes/No)? ").lower()
        print(" ")
```

Figure 2 - Add a new customer function part 1

If user enter 'Yes' as his/her answer, then all details added to the suitable **lists** using '**append**' function. Somehow account count reaches its' limit then that will show "Accounts Reach Maximum Amount !" and interface change to Main Manu. For this function to work **if** function was used. If user enter empty answers than "Something went wrong!" Will shoes and again give the option to try again or leave to Main Manu. For function to work, **try** and **except** keywords ware used from the bigging of the code.

```
        if (answer == str("yes")):
            cindex = len(custnum_list)
            if (cindex < 5):
                cust_count = 1
                accnum_list.append(bnk_acc_num)
                nic_list.append(nic)
                fname_list.append(f_name)
                lname_list.append(l_name)
                bdate_list.append(birth_date)
                padd_list.append(pmnt_add)
                phnum_list.append(ph_num)
                custnum_list.append(cust_count)
                bbalance_list.append(bank_balance)
                print("_____")
                print("        ""Account Added")
                print("_____")
                main()
            else:
                print("Accounts Reach Maximum Amount ! ")
                print("_____")
                main()
        else:
            main()
            print(" ")
    except ValueError:
        print("_____")
        print("Something went wrong!!.")
        print("_____")
        ansin = input("Do you want to Try Again (Yes/No)?")
        if (ansin == str("yes")):
            print("_____")
            customerinput()
        elif (ansin == str("no")):
            print("_____")
            main()
        else:
            print("_____")
            main()
```

Figure 3 - Add a new customer function part 2

Using “Do you want to Try Again (Yes/No)?” option for several numbers of time will increase the complexity of code but make it also a function was reducing the complexity. Using **def** keyword that was achieved. But this function only compatible with **add a new customer function only.**

```
def trycustomerin ():
    print(" ")
    print("You input WRONG charater count.")
    print("_____")
    ansinty = input("Do you want to Try Again (Yes/No)?").lower()
    if (ansinty == str("yes")):
        print("_____")
        customerinput()
    elif (ansinty == str("no")):
        print("_____")
        main()
    else:
        print("_____")
        main()
```

Figure 4 - Try again function for Add a new customer function

7.3. View customer detail function

To create a function for "View details of a customer" option, **def** keyword was used and name it as 'viewdetails' Then use **print** function to display the relevant details for interface. To view customer details, user need to give account number that stored in system. For that set a **variable** to get account number from user and with a combination of **len** function will find the correct position of a customer in **accnum_list**. After, using those value user can see all the details that relevant to the given account number. Finally, user need to answer, "Do you want to view another account details (Yes/No)?" If the user enters 'yes', then repeat this process again. If user enters 'no', then interface change to Main Manu. Somehow user enter nether 'yes' or 'no' then it gives "Account number not found." Arlet and again give option to user try again or not. For that function to work **try** and **except** keywords ware used.

```
def viewdetails ():                          # -2- view single customer detail function
    print(" ")
    print("                         ""ABC Bank")
    print("              ""View details of a customer ")
    print(" ")
    bnk_acc_num = int(input("Bank Account Number - "))
    print(" ")
    try:
        find_acc_num = accnum_list.index(bnk_acc_num)
        print("NIC                  - ", nic_list[find_acc_num])
        print("Phone Number         - ", phnum_list[find_acc_num])
        print("First Name           - ", fname_list[find_acc_num])
        print("Last Name            - ", lname_list[find_acc_num])
        print("Bank Balance         - ",'Rs.' ,bbalance_list[find_acc_num])
        print(" ")
        answer = input("Do you want to view another account details (Yes/No)? ").lower()
        if (answer == str("yes")):
            print("_____________________________________________________________________________")
            viewdetails()

        elif (answer== str("no")):
            print("_____________________________________________________________________________")
            main()
        else:
            print("_____________________________________________________________________________")
            print("Invalid input. Please enter Yes or No.")
            print("_____________________________________________________________________________")
            ansvdd = input("Do you want to Try Again (Yes/No)?").lower()
            if (ansvdd == str("yes")):
                print("_____________________________________________________________________________")
                viewdetails()
            elif (ansvdd == str("no")):
                print("_____________________________________________________________________________")
                main()
            else:
                print("_____________________________________________________________________________")
                main()

    except ValueError:
        print("_____________________________________________________________________________")
        print("Account number not found.")
        print("_____________________________________________________________________________")
        ansvd = input("Do you want to Try Again (Yes/No)?").lower()
        if (ansvd == str("yes")):
            print("_____________________________________________________________________________")
            viewdetails()
        elif (ansvd == str("no")):
            print("_____________________________________________________________________________")
            main()
        else:
            print("_____________________________________________________________________________")
            main()
```

Figure 5 - View details of a customer function

7.4. View all customers detail function

To create a function for “View details of all the customers” option, **def** keyword was used and name it as ‘viewall’ then use **print** function to display the relevant details for interface. Then set a variable to get a length(**len**) of **custnum_list** that holds the count of customers that registered in system. After that set **vd_count** as 0 and use **while** keyword to repeat the process. From the first time the code runs it will display Account number, NIC, First name, Last name and the bank balance. Then **vd_count** value assign to 1. After that code repeats until the condition fails. For final the user has an option to update account details, if user enter ‘yes’ then it will move to update function. If the user enters ‘no’ then interface moves to Main Manu. Somehow user enter other than ‘yes’ or ‘no’ then “Invalid answer !” alert will display and move to Main Manu.

```
def viewall ():                    # -3- View all customers details function
    print(" ")
    print("                        ""ABC Bank")
    print("              ""View Details of all the customers")
    print(" ")
    lindex = len(custnum_list)
    vd_count = 0
    while (vd_count < lindex):
        print("Account No.  -", accnum_list[vd_count])
        print("NIC          -", nic_list[vd_count])
        print("First Name   -", fname_list[vd_count])
        print("Last Name    -", lname_list[vd_count])
        print("Bank Balance -", 'Rs.', bbalance_list[vd_count])
        print(" ")
        vd_count = vd_count + 1
    answer = input("Do you want to update the account details (Yes/No)? ").lower()
    if (answer == 'yes'):
        print("_____________________________________________________________________________")
        update()
    elif (answer == 'no'):
        print("_____________________________________________________________________________")
        main()
    else:
        print("_____________________________________________________________________________")
        print("Invalid Answer!!")
        print("_____________________________________________________________________________")
        main()
```

Figure 6 - View all customers details function

7.5. Deposit function

To create a function for “Deposit Money to a given account” option, def keyword was used and name it as ‘deposit’ then use print function to display the relevant details for interface. Then set a variable to get Bank account number and another variable to get Deposit amount form the user. After that user need to answer, “Do you want to save (Yes/No)?” question. If user enter ‘yes’, then find the accounts’ position in accnum_list using index function using deposit_num. after that add relevant account balance to user added value and print “Your new balance - ******” . Finally set all created variables to 0. If user enter ‘no’, then interface moves to Main Manu. However, user enter other than ‘yes’ or ‘no’ then display a hint as “invalid input. Please input yes or no.” and gave a user to try again option. Somehow what if user enter invalid account number in first place, then interface shows “Account number not found.” And gave user to try again option.

```
def deposit ():                                 # -4- Deposit function
    print(" ")
    print("                         ""ABC Bank")
    print("              ""Deposit Money to a given account")
    print(" ")
    try:
        deposit_num = int(input("Bank Account Number - "))
        print(" ")
        dep_amount = float(input("Deposit Amount      - "))
        print(" ")
        answer = input("Do you want to save (Yes/No)?").lower()
        if (answer == str("yes")):

            finddep_num = accnum_list.index(deposit_num)
            updated_damount = bbalance_list[finddep_num] + dep_amount
            bbalance_list[finddep_num] = updated_damount
            print("_____________________________________________________________________")
            print("       ""Your new balance - ", 'Rs.', bbalance_list[finddep_num])
            deposit_num = 0
            dep_amount = 0
            updated_damount = 0
            print("_____________________________________________________________________")
            main()
        elif (answer == str("no")):
            print("_____________________________________________________________________")
            main()
        else:
            print("_____________________________________________________________________")
            print("Invalid input. Please input Yes or No.")
            print("_____________________________________________________________________")
            ansdd = input("Do you want to Try Again (Yes/No)?").lower()
            if (ansdd == str("yes")):
                print("_____________________________________________________________________")
                deposit()
            elif (ansdd == str("no")):
                print("_____________________________________________________________________")
                main()
            else:
                print("_____________________________________________________________________")
                main()
    except ValueError:
        print("_____________________________________________________________________")
        print("Account number not found.")
        print("_____________________________________________________________________")
        ansd=input("Do you want to Try Again (Yes/No)?").lower()
        if (ansd == str("yes")):
            print("_____________________________________________________________________")
            deposit()
        elif (ansd == str("no")):
            print("_____________________________________________________________________")
            main()
        else:
            print("_____________________________________________________________________")
            main()
```

Figure 7 - Deposit function

7.6. Withdraw function

To create a function for “Withdraw money from a given account” option, def keyword was used and name it as ‘withdraw’ then use print function to display the relevant details for interface. After that set a variables to get account number and withdraw amount from the user. After that user have a question that “Do you want to save (Yes/No)?”. If user enters ‘yes’, then variable set to find user account position in accnum_list. Then set a condition if customer bank balance less than withdraw amount, display “Insufficient Balance!” and gave the option to try again or back to Main Manu. Otherwise withdraw balance disease bank balance. And update the new amount to relevant bank balance and after set variables to 0. If the user enters ‘no’ to the save question, then the interface move to Main Manu. However, user enter other that ‘yes’ or ‘no’, it will display invalid input and give it a try again option. Somehow user enter invalid account number in first place, then interface shows “Account number not found.” And gave user to try again option.

```
def withdraw():                                    # -5- Withdraw function
    print(" ")
    print("                         ""ABC Bank")
    print("              ""Withdraw money from a given account")
    print(" ")
    try:
        withdraw_num = int(input("Bank Account Number - "))
        print(" ")
        with_amount = float(input("Withdraw Amount     - "))
        print(" ")
        answer = input("Do you want to save (Yes/No)?").lower()
        if (answer == str("yes")):
            findwithd_num = accnum_list.index(withdraw_num)
            if (with_amount > bbalance_list[findwithd_num]):
                print("_____________________________________________________________")
                print("        ""Insufficient Balance!!")
                print("_____________________________________________________________")
                answw = input("Do you want to Try Again (Yes/No)?").lower()
                if (answw == str("yes")):
                    print("_____________________________________________________________")
                    withdraw()
                elif (answw == str("no")):
                    print("_____________________________________________________________")
                    main()
                else:
                    print("_____________________________________________________________")
                    main()
            else:
                updated_wamount = bbalance_list[findwithd_num] - with_amount
                bbalance_list[findwithd_num] = updated_wamount
                print("_____________________________________________________________")
                print("        ""Your new balance - ", 'Rs.', bbalance_list[findwithd_num])
                withdraw_num = 0
                with_amount = 0
                updated_wamount = 0
                print("_____________________________________________________________")
                main()
        elif (answer == str("no")):
            print("_____________________________________________________________")
            main()

        else:
            print("Invalid input. Please input Yes or No.")
            print("_____________________________________________________________")
            answw = input("Do you want to Try Again (Yes/No)?").lower()
            if (answw == str("yes")):
                print("_____________________________________________________________")
                withdraw()
            elif (answw == str("no")):
                print("_____________________________________________________________")
                main()
            else:
                print("_____________________________________________________________")
                main()

    except ValueError:
        print("_____________________________________________________________")
        print("Account number not found.")
        print("_____________________________________________________________")
        answ = input("Do you want to Try Again (Yes/No)?").lower()
        if (answ == str("yes")):
            print("_____________________________________________________________")
            withdraw()
        elif (answ == str("no")):
            print("_____________________________________________________________")
            main()
        else:
            print("_____________________________________________________________")
            main()
```

Figure 8 - Withdraw function

7.7. Update customer detail function

To create a function for “Update Customer Details” option, def keyword was used and name it as ‘withdraw’ then use print function to display the relevant details for interface. After that set a variable to get account number form user and check it character count equal to required amount, then turn integer data type into String and find account position in accnum_list using index function. After that one by one all new NIC numbers, First name, Last name, Birth date, permanent address and phone number can input to system if this all justify its’ character count as requested. After that interface give option (Do you want to save the new details (Yes/No)?) to answer. If somehow user enter wrong character count to any one, interface give the hint for required character count and try again option but user need to type again all details from the bigging.

```
def update ():                                  # -6- Details update function
    print(" ")
    print("                        ""ABC Bank")
    print("                    ""Update Customer Details")
    print(" ")
    try:
        bnk_acc_num = int(input("Bank Account Number    - "))
        test = str(bnk_acc_num)
        if (len(test) == 10):
            print(" ")
            find_acc_num = accnum_list.index(bnk_acc_num)
            n_nic = str(input("NIC                      - "))
            test = str(n_nic)
            if (len(test) == 10):
                n_f_name = input("First Name             - ")
                test = str(n_f_name)
                if (len(test) < 11):
                    n_l_name = input("Last Name              - ")
                    test = str(n_l_name)
                    if (len(test) < 16):
                        n_birth_date = str(input("Birth Date             - "))
                        test = str(n_birth_date)
                        if (len(test) < 18):
                            n_pmnt_add = input("Permanent Address      - ")
                            test = str(n_pmnt_add)
                            if (len(test) < 16):
                                n_ph_num = str(input("Phone Number           - "))
                                test = str(n_ph_num)
                                if (len(test) == 10):
                                    print(" ")
                                    answer = input("Do you want to save the new details (Yes/No)? ").lower()
                                else:
                                    print("_____________________________________________________________")
                                    print("Phone Number must be 10 Digit.")
                                    trycustomerup()
                            else:
                                print("_____________________________________________________________")
                                print("Permanent Address maximum of 15 characters.")
                                trycustomerup()
                        else:
                            trycustomerup()
                    else:
                        print("_____________________________________________________________")
                        print("Last Name maximum of 15 characters.")
                        trycustomerup()
                else:
                    print("_____________________________________________________________")
                    print("First Name maximum of 10 characters.")
                    trycustomerup()
            else:
                print("_____________________________________________________________")
                print("NIC must be 10 Digit.")
                trycustomerup()
        else:
            print("_____________________________________________________________")
            print("Bank Account number must be 10 Digit.")
            trycustomerup()
```

Figure 9 - Customer details update function part 1

If user enter answer as ‘yes’, then all new details add to the relevant account number and displays “Account Updated”. If the user enters ‘no’ then the interface moves to Main Manu. However, the user enters something other than ‘yes’ or ‘no’, then displays “invalid input. Please enter Yes of No” and give try again option to re try or leave to Main Manu.

```
        if (answer == str("yes")):
            nic_list[find_acc_num] = n_nic
            fname_list[find_acc_num] = n_f_name
            lname_list[find_acc_num] = n_l_name
            bdate_list[find_acc_num] = n_birth_date
            padd_list[find_acc_num] = n_pmnt_add
            phnum_list[find_acc_num] = n_ph_num
            print("_____")
            print("       ""Account Updated")
            print("_____")
            main()
        elif answer == "no":
            print("_____")
            main()
        else:
            print("_____")
            print("Invalid input. Please enter Yes or No.")
            print("_____")
            ansu = input("Do you want to Try Again (Yes/No)?").lower()
            if (ansu == str("yes")):
                print("_____")
                update()
            elif (ansu == str("no")):
                print("_____")
                main()
            else:
                print("_____")
                main()

    except ValueError:
        print("_____")
        print("something went wrong!!")
        print("_____")
        ansup = input("Do you want to Try Again (Yes/No)?").lower()
        if (ansup == str("yes")):
            print("_____")
            update()
        elif (ansup == str("no")):
            print("_____")
            main()
        else:
            print("_____")
            main()
        update()
```

Figure 10 - Customer details update function part 2

For this occasion, try again function use more time than usual, so adding every try again option for code would increase its’ complexity. For that reason, using **def** keyword build a function called trycustomerup and add it to update function. But this function is compatible with **update function** only.

```
# ---------------------------------SPECIAL FUNCTION FOR CUSTOMER detail update-----------------------------
def trycustomerup ():
    print(" ")
    print("You input WRONG charater count.")
    print("_____")
    ansupty = input("Do you want to Try Again (Yes/No)?").lower()
    if (ansupty == str("yes")):
        print("_____")
        update()
    elif (ansupty == str("no")):
        print("_____")
        main()
    else:
        print("_____")
        main()
```

Figure 11 - Try again function for customer details update function

7.8. Main Manu function

After creating all functions that system required, create a function called Main Manu using def keyword. And using print function display relevant details and all the options that required to build for system including Exit function. After that set an integer datatype variable as yout_chice to get user selection 1 – 7. Set a condition if user enter 1, then it will lead to **customerinput** function, or if user enter 2, then it will lead to **viewdetails** function, or if user enter 3, then it will lead to **viewall** function, or if user enter 4, then it will lead to **deposit** function, or if user enter 5, then it will lead to **withdraw** function, or if user enter 6, then it will lead to **update** function and if user enter 7, then program ask to stop the program after displays "Good Day!!!". Somehow if user enter less than 1 or more than 7, then displays "Out of choice range, Try again!!" and back to Main Manu and if user enter empty the program also come to Main Manu. After creating all functions that need to run program correctly, placed Main Manu function at the end to start the program.

```
def main():                                    # main manu function
    try:
        print(" ")
        print("                        ""ABC Bank")
        print("                       ""Main Manu")
        print(" ")
        print("1) Add a new customer")
        print("2) View details of a customer including his/her bank balance")
        print("3) View details of all the customers with their bank balances")
        print("4) Deposit money to a given account")
        print("5) Withdraw money from a given account")
        print("6) Update Customer Details")
        print("7) Exit")
        print(" ")
        your_choice = int(input("                                              ""Your Choice : "))

        print("_____________________________________________________________________________")

        # 1-add a new customer
        if (your_choice == 1):
            customerinput()

        # 2-View details of a customer including his/her bank balance
        elif (your_choice == 2):
            viewdetails()

        # 3-View details of all the customers with their bank balances
        elif (your_choice == 3):
            viewall()

        # 4-Deposit Money to a given account
        elif (your_choice == 4):
            deposit()

        # 5 -Withdraw money from a given account
        elif (your_choice == 5):
            withdraw()

        # 6-Update Customer Details
        elif (your_choice == 6):
            update()

        # 7 -Exit
        elif (your_choice == 7):
            print("Good Day!!!")
            print("_____________________________________________________________________________")
            exit()

        # if more than 7
        elif (your_choice > 7):
            print("Out of choice range, Try again!!")
            print("_____________________________________________________________________________")
            main()
        # if less than 1
        elif (your_choice < 1):
            print("Out of choice range, Try again!!")
            print("_____________________________________________________________________________")
            main()
    except ValueError:
        print("_____________________________________________________________________________")
        main()

main()
```

Figure 12 - Main Manu function

8. Algorithm which you have taken to approach the solution.

1. Start.
2. Display the 7 options for the user.
   2.1. Add a new customer.
   2.2. View details of a customer including his/her bank balance.
   2.3. View details of all the customers with their bank balances.
   2.4. Deposit money to a given account.
   2.5. Withdraw money from a given account.
   2.6. Update Customer Details.
   2.7. Exit.
3. Handle the user's choice.
   3.1. If user choose choice 1 (Add a new customer)
      3.1.1. Prompt the user to input all details of a customer one by one.
      3.1.2. Display an option as "Do you want to save the account (Yes/No)?"
         3.1.2.1. If user enter 'yes', Display "Account Added"
         3.1.2.2. If user enter 'No', Display Main Manu
      3.1.3. If user input wrong character type or invalid character count, display an error message with hint to correct next time and give option to try again or to go to Main Manu
   3.2. If user choose choice 2 (View details of a customer including his/her bank balance.)
      3.2.1. Prompt the user to input account number.
         3.2.1.1. If user enter valid account number, display details that relevant to that account number and give option as Do you want to view another account details (Yes/No)?
            3.2.1.1.1. If user enter 'yes', again user get the chance to enter account another account number again
            3.2.1.1.2. If user enter 'no', display Main Manu
         3.2.1.2. If user enter invalid account number. Then user get error message and try again option to enter account number or ga back to Main Manu
   3.3. **If the user choose choice 3 (View details of all the customers with their bank** balances.)

3.3.1. Display all the users' details that registered in system.

3.3.2. Display Do you want to update the account details (Yes/No)?

3.3.2.1. If user enter 'Yes', display "Update Customer Details." Interface.

3.3.2.2. If user enter 'No', display Main Manu.

3.4. If user choose choice 4 (Deposit money to a given account.)

3.4.1. Prompt the user to input account number.

3.4.2. Display prompt to Deposit Amount .

3.4.3. Display "Do you want to save (Yes/No)?"

3.4.3.1. If user enter 'Yes',

3.4.3.1.1. If it valid account number, display "Your new balance"

3.4.3.1.2. If it invalid account number, display "Account number not found." And try again option

3.4.3.2. If user enter 'No', go back to Main Manu

3.5. If user choose choice 5 (Withdraw money from a given account.)

3.5.1. Prompt the user to input account number.

3.5.2. Display prompt to Withdraw Amount .

3.5.3. Display "Do you want to save (Yes/No)?"

3.5.3.1. If user enter 'Yes',

3.5.3.1.1. If it a valid account number and customer bank balance more than withdraw amount, display "Your new balance"

3.5.3.1.2. If it a valid account number and customer bank balance less than withdraw amount, display "Insufficient Balance!!" and try again option.

3.5.3.1.3. If it a invalid account number, display "Account number not found." And try again option.

3.5.3.2. If user enter 'No', display Main Manu.

3.6. If user choose choice 6 (Update Customer Details.)

3.6.1. Prompt the user to input account number.

3.6.2. If it's a valid account number, display Prompt to input all new details of a customer one by one

3.6.2.1. If all character types and counts are correct, display option as "Do you want to save the new details (Yes/No)?"

3.6.2.1.1. If user enter 'Yes', display "Account Updated"

3.6.2.1.2. If user enter 'no', display option to try again or go to Main Manu.

3.6.2.2. If character types or counts are incorrect, display an error message with hint to correct next time and give option to try again or to go to Main Manu

3.6.3. If it's an invalid account number, also display an error message with hint to correct next time and give option to try again or to go to Main Manu.

3.7. If user choose choice 7(Exit.)

3.7.1. Display "Good Day!!!"

3.7.2. Stop the program.

9. Problems and solutions

1. How to move across the algorithm:

   To solve this problem, I used **def** keyword to create functions to each option. That allows me to build an algorithm that can move any option without getting crashed.

```
def trycustomerup ():
```

2. How to set character limit:

   To solve this problem, I used to turn all input data types into **string** and then use **len** function to get character count and set a limit as condition using **if** keyword.

```
nic = str(input("NIC -"))
test = str(nic)
if (len(test) == 10):
    print("Go forward")
else:
    print("NIC must be 10 digits.")
```

3. How to store data in properly:

   To solve this problem, I used **lists**. It helps to store data without getting mix with others using **append** function.

```
nic_list=[]
nic=int(input("NIC -"))
nic_list.append(nic)
```

4. How to find correct account form account number:

   To solve this problem, I used **index** function to find account position with the help of account number when needed.

```
acc_num=int(input("Account No.-"))
find_acc=accnum_list.index(acc_num)
```

5. How to set account count limit:

   To solve this problem, I used condition using **if** keyword. First, I get length of custnum_list using **len** function and set a condition.

```
cindex = len(custnum_list)
if (cindex < 5):
    print("You good")
else:
    print("Accounts Reach Maximum Amount")
```

6. How to avoid crashing when user enter invalid input:

   To solve this problem, I used **try** and **except** keywords. I put try in begging of all algorithms and except at the end. And I put try again function to restart algorithm again or change direction to other function.

```
try:
    answer=input("Yes/No")
    if (answer == str("Yes")):
        print("OK")
    else:(answer == str("No")):
        print("OK")
except ValueError:
    print("Try Again")
```

10. Test cases

10.1. Case 1

Choice	Figure	Input	Expected Output	Actual output	result
1	1	Input all details with Right data types and correct character amounts. 1. Account Number (10-digit number) 2. NIC (String, with 10 characters) 3. First Name (String, maximum of 10 characters) 4. Last Name (String, maximum of 15 characters) 5. Birth Date 6. Permanent Address (String, maximum of 15 characters) 7. Telephone Number (String, 10 characters)	Successfully added to relevant lists in background and Display "Account Added"	Display "Account Added"	pass
1	2	Input details with wrong data type	Display Error message and Display Try again option	Display "Something went wrong!." and Display "Do you want to Try Again (Yes/No)?"	pass
1	3	Input details with wrong character count	Display relevant character count and error message. Display try again option.	Display relevant character count and error message. Display "Do you want to Try Again (Yes/No)?"	pass
1	4	Input 6^{th} account	Display error massage and go to Main Manu	Display 'Accounts Reach Maximum Amount !"	pass

Table 1 - Test case 1

```
                    ABC Bank
                Add a new customer

Bank Account Number       - 1231231239
NIC                       - 234234234v
First Name                - John
Last Name                 - Perera
Birth Date                - 1980/12/12
Permanent Address         - No.30 Galle
Phone Number              - 0771111111

Do you want to save the account (Yes/No)? yes


       Account Added
```

Test case C1 - figure 1

```
                    ABC Bank
                Add a new customer

Bank Account Number       - sss

Something went wrong!.

Do you want to Try Again (Yes/No)?yes
```

Test case C1 - figure 2

```
                    ABC Bank
                Add a new customer

Bank Account Number       - 1234567890
NIC                       - 111222

NIC must be 10 digits.

You input WRONG charater count.

Do you want to Try Again (Yes/No)?
```

Test case C1 - figure 3

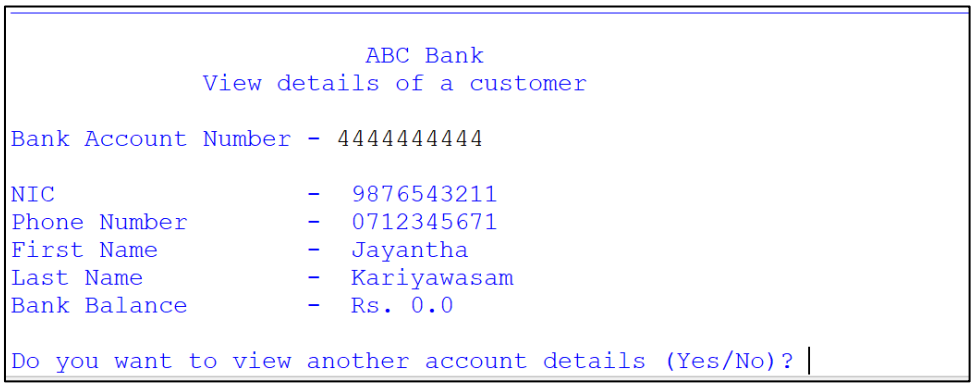
```
                    ABC Bank
                Add a new customer

Bank Account Number       - 6666666666
NIC                       - 6666666666
First Name                - Edirlweera
Last Name                 - Sarath
Birth Date                - 1965/06/17
Permanent Address         - Horaupathana
Phone Number              - 0912234511

Do you want to save the account (Yes/No)? yes

Accounts Reach Maximum Amount !

                    ABC Bank
                   Main Manu

1) Add a new customer
2) View details of a customer including his/her bank balance
3) View details of all the customers with their bank balances
4) Deposit money to a given account
5) Withdraw money from a given account
6) Update Customer Details
7) Exit

                                        Your Choice :
```

Test case C1 - figure 4

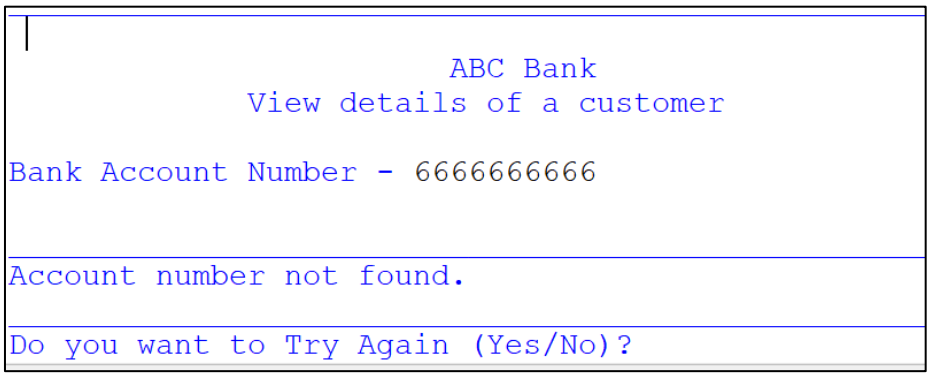
10.2. Case 2

Choice	Figure	Input	Expected Output	Actual output	result
2	1	Input valid bank account number	Display details relevant to that Account number and display “Do you want to view another account details”	Display details that relevant to the Account number and display “Do you want to view another account details”	pass
2	2	Input invalid bank account number	Display error message and try again .	Display “Account number not found.” And “Do you want to Try Again (Yes/No)?”	pass

Table 2 - Test case 2



Test case C2 - Figure 1

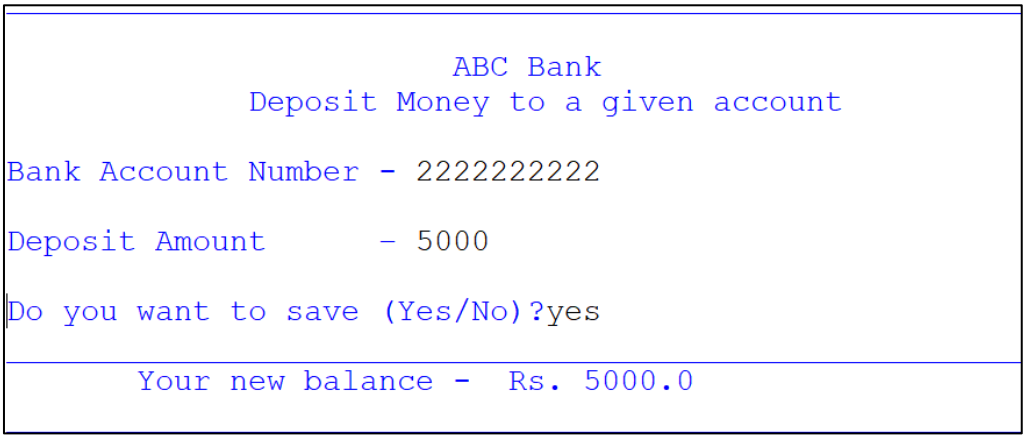


Test case C2 - Figure 2

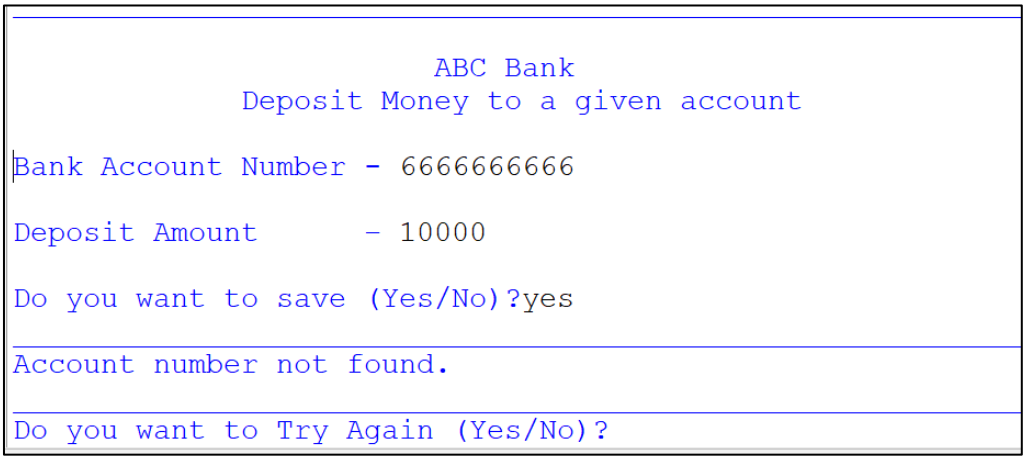
10.3. Case 3

Choice	Figure	Input	Expected Output	Actual output	result
4	1	Enter a valid bank account number and enter the Deposit amount. Then – save	New balance	Your new account balance	pass
4	2	Enter invalid bank account number and enter the Deposit amount. Then – save	Error message	Display "Account number not found.' And try again option.	pass

Table 3 - Test case 3



Test case C3 - Figure 1

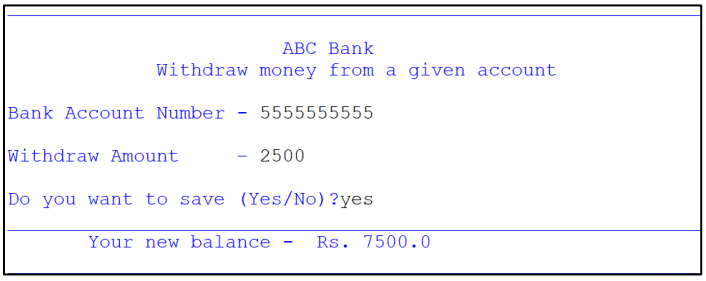


Test case C3 - Figure 2

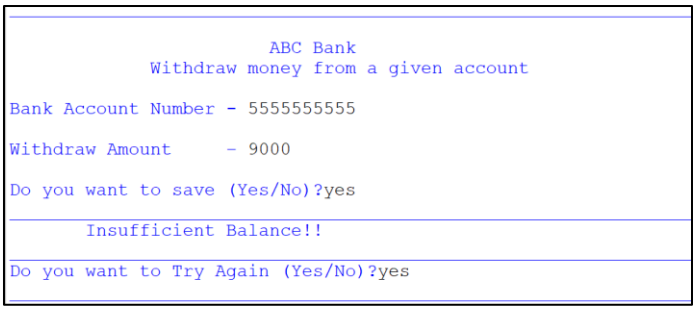
10.4. Case 4

Choice	Figure	Input	Expected Output	Actual output	result
5	1	Enter a valid bank account number and enter the valid withdraw amount. Then – save	New balance	Your new account balance	pass
5	2	Enter a valid bank account number but invalid withdraw amount Then - save	Error message	Display “Insufficient Balance!!” and try again option.	pass
5	3	Enter invalid bank account number	Error message	Display “something went wrong!!” and try again option	pass

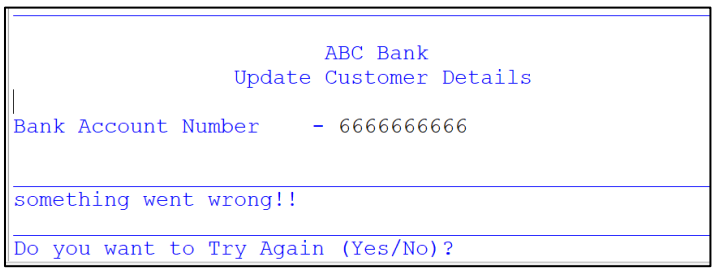
Table 4 - Test case 4



Test case C4 - Figure 1



Test case C4 - Figure 2

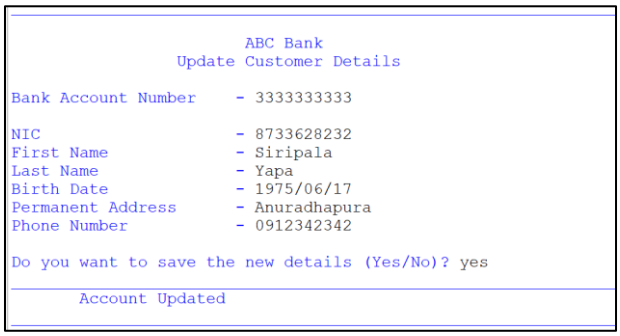


Test case C4 - Figure 3

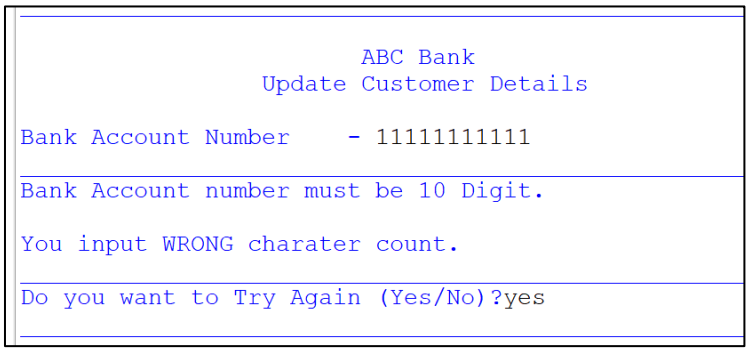
10.5. Case 5

Choice	Figure	Input	Expected Output	Actual output	result
6	1	Enter correct Account number and add details with correct data types and correct character count	Customer details update in the background and display “Account Updated”	Display “Account Updated”	pass
6	2	Enter the wrong character count (more 10 or less than 10)	Hint and error message with try again option	Display “Bank Account number must be 10 Digit.” And “You input wrong character count.” And Try again option.	pass
6	3	Enter the right account number but wrong data type or the wrong character count for details	Hint, Error and try again option	Display “First Name maximum of 10 characters. And You input wrong character count.” try again option	pass

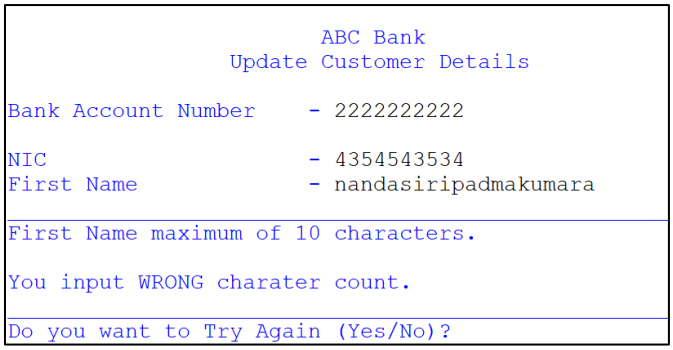
Table 5 - Test case 5



Test case C5 - Figure 1



Test case C5 - Figure 2



Test case C5 - Figure 3

10.6. Case 6

Choice	Figure	Input	Expected Output	Actual output	result
3	1	Nothing to input	All customer details that registered. And display “Do you want to update the account details”	All customer details that registered. And display “Do you want to update the account details”	pass

Table 6 - Test case 6

Test case C6 - Figure 1