

7. Utilizing 'functions' concepts in python programming.

10/9/25

a. Banking Transaction system.

Aim:- To develop a python program using functions that simulates basic banking transactions: deposit, withdraw, and checking the account.

Algorithms

1. Initialize account balance to zero.
2. Define a function to deposit money, which increases the balance.
3. Define a function to withdraw money, checking if the balance is sufficient.
4. Define a function to display the current balance.
5. Use menu-driven options to perform deposit, withdraw, and balance check actions.

Python program

balance = 0

```
def deposit(amount):  
    global balance  
    balance += amount  
    print("Deposited:", amount)
```

```
def withdraw(amount):  
    global balance  
    if amount <= balance:  
        balance -= amount  
        print("withdrawn:", amount)  
    else:  
        print("Insufficient Balance")
```

```
def check_balance():  
    print("Current Balance:", balance)
```

Output: ~~Current balance is 300.~~

Deposited: 500

Withdrawn: 200

Current Balance: 300

Insufficient Balance.

Current Balance: 300

Example usage.

```
deposit(500)  
withdraw(200)  
check-Balance()  
withdraw(400)  
check-balance()
```

00

~~Deposited: 500
Withdrawn: 200
Current Balance: 300
Insufficient Balance.
Current Balance: 300~~

Result:-

The program performs banking transactions using functions and maintains the account balance accurately.

b. Student Result calculator

Aim:-

To create a python program using functions to accept marks of three subjects, calculate total, average, grade and display.

Algorithm

1. Define a function to accept marks for three subjects.
2. Define a function to calculate the total and average.
3. Define a function to determine the grade. (A/B/C/fail) based on average.
4. Define a separate function to display the result.

Python program

```
def accept_marks():
    m1 = int(input("Enter marks for subject 1: "))
    m2 = int(input("Enter marks for subject 2: "))
    m3 = int(input("Enter marks for subject 3: "))
    return m1, m2, m3

def calculate_result(m1, m2, m3):
    total = m1 + m2 + m3
    average = total / 3
    if average >= 75:
        grade = 'A'
    elif average >= 60:
        grade = 'B'
    elif average >= 40:
        grade = 'C'
    else:
        grade = 'fail'
    return total, average, grade.
```

Output

Enter marks for subject 1: 80

Enter marks for subject 2: 70

Enter marks for subject 3: 60

Total marks: 210

Average Marks: 70.0

Grade: B

```
def display_result(total, average, grade):  
    print("Total Marks:", total)  
    print("Average Marks:", average)  
    print("Grade:", grade)
```

marks = accept_marks()

total, average, grade = calculate_result(marks)
display_result(total, average, grade).

VEL TECH - CSE	
EX NO.	7-
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	1
TOTAL (20)	15
SIGN WITH DATE	8/3/25

Result: - The program uses functions to process student marks and displays a result including total, average, and grade classification.