13/08/25

Task 4:- Use various data types, List types, and Dictionary in Python. Programming.

(a) Shopping Cart Price Calculator (List)

Aim:- To store item prices in a list and calculate the total bill amount, highest-priced item, and lowest-priced item.

Algorithm:-

1. start
2. Create a list to store the prices of items.
3. Calculate the total bill using sum().
4. Find the highest price using max().
5. Find the lowest price using min().
6. Display the results.
7. End.

Program:-

```
Prices = [120, 85, 300, 150, 60]
Total_bill = sum(Prices)
highest_price = max(Prices)
lowest_price = min(Prices)
Print("Total Bill Amount:", total_bill)
Print("Highest Priced Item:", highest_price)
Print("Lowest Priced Item:", lowest_price)
```

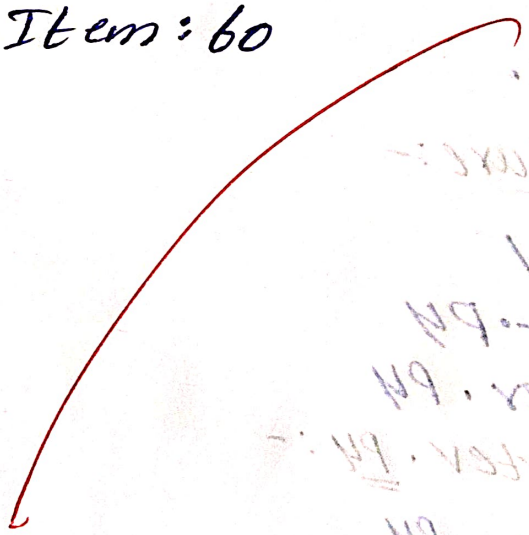
Result:- The Program successfully calculated the total bill, Highest Priced item, and lowest-priced item from the Given list of Prices.

Output:

Total Bill Amount : 715

Highest Priced Item: 300

lowest priced Item : 60



(6) Student Exam Result (TUPLE)

Aim:- To store each student's name and marks in a tuple and display the student with the highest marks and students who scored above 400.

Algorithm:-

1. Start
2. Store student name and marks in a list of tuples.
3. Find the student with the highest marks using max() with key function.
4. Loop through all students and display those with marks above 400.
5. End.

Program:-

```
students = [
    ("Rahul", 456),
    ["Priya", 390),
    ("Karan", 420),
    ("Anita", 470),
    ("Vijay", 365)
]
highest_student = max(students, key = lambda x: x[1])
Print ("Student with Highest Marks:", highest_student[0],
       "_", highest_student [1])
Print ["Students scoring above 400!")
for name, marks in students:
    if marks > 400:
    Print (name, "_", marks)
```

output:-

student with Highest Marks: Anita — 470

student scaring above 400:

Rahul — 456

karan — 420

Anita — 470

Result:-

The program successfully identified the top scorer and listed all students with marks above 400.

(c) Country - Capital Finder (Dictionary)

Aim:-

To store country-capital pairs in a dictionary, add a new pair, search for a capital, and Display Pairs in alphabetical order..

Algorithm:-

1. Start
2. Create a dictionary with same country - capital pairs
3. Ask user to add a new country - capital pair.
4. Search for the capital of a country given by the user.
5. Display all country, capital pairs in alphabetical order.
6. End.

Program:-

```
countries = {
    "India": "New Delhi",
    "France": "paris",
    "Japan": "Tokyo"
}
new_country = input("Enter a new country:")
new_capital = input("Enter its capital:")
countries[new_country] = new_capital
search_country = input("Enter country to find its capital:")
if search_country in countries:
    Print("Capital of", Search_country, "is",
          countries[search_country])
else:
    Print("country not found!")
Print("\n country - capital Paris in alphabetical creator:")
```

Output:-

Enter a new Country : UK

Enter its capital : London

Enter country to find its capital : Japa

capital of Japan is Tokyo

Country — capital pairs in alphabetical o

France — Pairs

India — New Delhi

Japan — Tokyo

UK — London

```
for country in sorted (countries):
    Print (country, "—", countries[country])
```

VEL TECH (A/U)	
EX NO.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	25
SIGN WITH DATE	

20/8/25

Result:-

The Program successfully stored, updated, searched, and displayed country — capital pairs in alphabetical order.

20/8/25