

13/8/20 Task 4. Use various data types, list, tuples and

(a). dictionary in python programming

Aims- To calculate the total bill amount, highest-priced item, and lowest - priced item from a list of purchased items.

1. Start the program.
2. Create a list to store item prices.
3. Calculate the total bill using sum().
4. Find the highest price using max().
5. Find the lowest price using min().
6. Display the total, highest, and lowest prices.
7. End program.

Program

```
# Shopping cart price calculator using list
```

```
prices = [120, 250, 75, 300, 150] # sample prices
```

```
total = sum(prices)
```

```
highest = max(prices) # highest price
```

```
lowest = min(prices) # lowest price
```

```
print("Prices of items:", prices)
```

```
print("Total Bill Amount:", total)
```

```
print("Highest priced item:", highest)
```

```
print("Lowest priced item:", lowest)
```

Result:

The program successfully calculates the total bill, highest and lowest prices items using a list.

Sample output -

Prices of items: [120, 250, 75, 300, 150] and

Total bill amount = 895

Highest priced item: 300

lowest priced item: 75

AVERAGE - CSB	
EX NO.	3
PERFORMANCE (%)	95
RESULTS AND ANSWERS (%)	95
VIVA VOCES (%)	95
REGARD (%)	95
TOTAL (%)	95

(b) student exam result (tuple)

Aim: To store student names with their marks using tuples and display the highest scorer and students scoring above 400.

Algorithm:

1. Start the program
2. store student data as a list of tuples (name,marks).
3. find the student with the highest marks using max()
4. loop through tuples to display students scoring above 400.
5. Display the results.
6. End program

Program:

```
#Student exam result using Tuples.
```

```
Students = [ ("Rahul", 456), ("Anita", 389), ("Kiran", 412),  
            ("Sneha", 478), ("Arjun", 365)]
```

```
Highest = max(Students, key=lambda x: x[1])
```

```
Print("student Records: ", Students)
```

```
Print("Topper = ", Highest[0], "with", Highest[1], "marks")
```

```
Print("Students scoring above 400: ")
```

```
for name, marks in Students:
```

```
    if marks > 400:
```

```
        Print(name, ":", marks)
```

```
    print("Total students above 400 marks: ", count)
```

```
    count = count + 1
```

Results

The program correctly finds the highest scorer and lists students who scored above 400 marks.

student's name in Student table
and stores his date of birth in Date table.
A student having a name and birth date,
will be stored in Student table.

and price of books will also be stored.
A book price will be stored in Book table.
Cinema price along with its name
will be stored in Cinema table.

Sample output

Student Records: [('Rahul', 456), ('Anita', 389), ('Kiron', 412),
('Sneha', 478), ('Arjun', 36)]

Topper = Sneha with 478 marks (with "mark" to "marks") from
students scoring (above 400) (with "score" being "marks") from

Rahul = 456

Kiron = 412

Sneha - 478.

Entered in database. All records are stored in
the system with having stored in a selected file.

Program

Python

country - capital finder (Dictionary)

countries = { "India": "New Delhi", "France": "Paris",
"Tokyo": "Tokyo" }

Add new entry

Country = input ("Enter a country: ")

Capital = input ("Enter its capital: ")

countries[Country] = Capital

search for a capital

search_country = input ("Enter country to search: ")

if search_country in countries:

print ("Capital of " + search_country + " is")

(countries[search_country])

else:

print ("Country not found")

print ("All country - capital pairs (sorted): ")

for c in sorted (countries.keys()):

print (c, ":", countries[c])

Sample output

Enter a country: Germany

Enter its capital: Berlin

Enter country to search Capital: India

Capital of India is New Delhi

All country - capital pairs (sorted):

France : Paris

Germany : Berlin

India : New Delhi

Japan : Tokyo.

c. Country-Capital finder (Dictionary)

Aim:- To stores countries and capitals in dictionary and perform operations: add new entry, search for capital, display all pairs alphabetically.

Algorithm

1. Start the program.
2. Create a dictionary with country-capital pairs.
3. Add a new country-capital pair entered by user.
4. Search for a country's capital using dictionary lookup.
5. Display all country-capital pairs sorted by country name.
6. End the program.

[("Spain", "Madrid"), ("USA", "Washington"), ("UK", "London"),
("France", "Paris"), ("Germany", "Berlin"), ("Italy", "Rome"),
("Australia", "Canberra"), ("New Zealand", "Wellington")]

VEL TECH - CSE

EX NO.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	15

Result:- The program successfully adds, searches, and displays country-capital pairs using a dictionary.