

Date:- 13/8/25

Task 4:- Use various data types / list, tuple and dictionary in python programming.

a) shopping cart price = calculator (list)

Aim:- To store prices of purchased items in a list and calculate.

- Total bill amount.
- Highest Price item
- Lowest Price item

algorithm:

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

Program:-

Shopping cart price calculator

prices = [250, 120, 75, 300, 150]

total_bill = sum(prices)

Highest price = max(prices)

lowest price = min(prices)

Sample output

to far (Star B.C) highest = 845/- lowest
middle broad range priced item = 360/-
highest priced item = 25/-
lowest priced item = 15/-

~~✓ Edwards 118 118
✓ Mrs. 22,149 118
✓ Mrs. 62,149 118~~

Handwritten

and wife wrote off from
Cross Brew 11th Inst 2500.
Brew Inst. 22nd Inst 2500.
Food off 1000.
(Cross Brew Inst 2500))
(Inst 2500))
off 1000 off 1000
Cross Brew 11th Inst 2500.
Food off 1000.
Inst 2500
Cross Brew 11th Inst 2500.
Food off 1000.

print ("Total Bill Amount:", total_bill)

print ("Highest Priced Item:", highest_price)

print ("Lowest Priced Item:", lowest_price)

b) Student Exam Result (tuple)

Aim is to store student 'name' and marks in tuples, then:

Display student with highest marks.

Display student scoring above 400.

Algorithm:-

1. Start Program

2. Store data for 5 student on a list of tuples.

3. Find the student with highest marks using max() and a they function.

4. Loop through list to display students with marks > 400

5. End Program.

Program

Student Exam Result

student = [

("Rahul", 456),

("Sneha", 389),

("Amit", 412),

("Preya", 478),

("Karan", 390)

]

Sample Output

top student: priya with 478 Marks

Students scoring above 40% are

Rahul = 456

Amit = 412

Priya = 478

So far no trouble yet what about

new stem loaded there should be no trouble yet but if you add 100 to the value of 100 it becomes 200 which is extreme

margin too

Aug 31 marks trouble

B = trouble

(222 "Lufot")

(100 "ordind")

```
top - student = max (students, top: lambda: [i])  
print ("Top student : " ; top - student [0], with  
      , top - student [i], "marks")
```

print ("Student scoring above 400: ")

for name, marks in students

if marks > 400 = print (name, "-"; marks)

C. country - capital Finder (Dictionary)

Aim:- To store countries and their capitals in a dictionary allowing for insertion, deletion, search.

- Adding a new entry
- Searching for a capital
- Displaying all pairs alphabetically by country name.

Algorithm:-

- ```
→ Start program
→ Create dictionary of countries and capitals
→ Take user input - for new country & capital pair and add to dictionary
→ Ask user for a country to search and display its capital.
→ Display all country - capital pairs sorted alphabetically.
→ End program.
```

### Program

# country - capital Finder

countries = { "India": "New Delhi", "France":

"Paris", "Japan": "Tokyo" }

(you enter the country)

single output

enters country name ~~and~~ enters name of

enter its capital = Berlin

enter country to search: Japan

capital of Japan is Tokyo

All country - capital pairs  
(Alphabetical):

France - Paris

Australia - Berlin

India - New Delhi

Japan - Tokyo

China - Beijing

Morocco

President

USA

China

Japan

China

Japan

China

# Add new entry

new-country = input("Enter country name: ")

new-capital = input("Enter capital of country: ")

countries[new-country] = new-capital

# Search for 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.")

# Display sorted list

print("In All country - capital Pairs (Alphabetical,")

for country in sorted(countries):

print(country, ":",

| EX NO          | Countries (country) |
|----------------|---------------------|
| PER 02/04/2023 | 51                  |
| RESULT         | 5                   |
| VIVA VOICE     | 2                   |
| RECORD         | 26                  |
| TOTAL (20)     | 25                  |
| SIG            |                     |

Resulting thus, various datatype, list,  
tuple and dictionary in python  
programming was used and verified  
successfully.