

4. /13/08/25 Use the various data types, list, tuples and dictionary in python programming.

15  
Q3  
Ans

# Task-4: Use various data types, List, Tuples And Dictionary In Python

## (a) Shopping Cart Price Calculator (List)

### Aim:-

To create a python program that uses a list to store the prices and calculates the total bill amount, the highest priced item, and the lowest priced item.

### Algorithm:-

1. Initialize an empty list to store item prices
2. Prompt the user to enter item prices and add them to the list. Continue until the user indicates they are done.
3. Calculate the sum of all prices in the list to get the total bill.
4. Find the maximum value in the list to get the highest priced item.
5. Find the minimum value in the list to get the lowest priced item.
6. Display the total bill, highest priced item, and lowest priced item.

### Program:-

```
prices []
```

```
while True:
```

```
    price_input = input("Enter item price (or 'done' to finish):")
```

```
    if price_input.lower() == 'done':
```

```
        break
```

```
    try:
```

```
        price = float(price_input)
```

```
        prices.append(price)
```

```
    except ValueError:
```

```
        print("Invalid input. Please enter a number or 'done'.")
```

```
if prices:
```

```
    total_bill = sum(prices)
```

```
    highest_price = max(prices)
```

```
    lowest_price = min(prices)
```

Output of the program is as follows:

modified GE program is

(HIS) modified 3209 420

DATA

It has a main function reading the file  
that contains the sales details from a file and  
process it from which it takes the first value as  
the total number of items.

Now we have to calculate the total price and quantity  
of the bill along with the item and their  
details are as follows:

For the first item quantity is 100 and total price is 1000.

### Output:

Total Bill Amount : 715

Highest Priced Item : 300

Lowest priced Item : 60

(b) Output of the program is as follows:

```
print(f "In Total Bill Amount: ${total_bill:.2f}\")\nprint(f "Highest priced item: ${highest_price:.2f}\")\nprint(f "Lowest priced item: ${lowest_price:.2f}\")\nelse:\n    print("No prices entered.")
```

### Result :-

The program successfully calculates the total bill, highest, and lowest priced items from a list of user-entered prices.

### 3/8/25 (b) student Exam Result (Tuple)

#### Aim:-

To create a python program that stores student names and marks in tuples, displays the student with the highest marks, and lists all students who scored above 400 marks.

#### Algorithm:-

1. Start the program
2. Initialize an empty list to store student tuples (name, marks).
3. Loop 5 times to get input for 5 students:
  - a. prompt for student name and marks.
  - b. Create a tuple (name, marks).
  - c. Add the tuple to the list.
4. Find the student with the highest marks by iterating through the list and comparing marks.
5. Iterate through the list again to identify and display students who scored above 400 marks.
6. Display the student with the highest marks and the list of students scoring above 400 marks.
7. Stop the program.

#### Program:-

```
students = []
for i in range(5):
    name = input("Enter name of student {i+1}: ")
    while True:
        try:
            marks = int(input("Enter marks for {name}: "))
            if marks < 0:
                print("Marks cannot be negative. Please enter a valid number.")
            else:
                break
        except ValueError:
            print("Invalid input. Please enter a number for marks.")
    students.append((name, marks))
```

(contd) (part) year 3 2021-22

student students needs both company staff for the students  
and other teachers with a particular subject in the school  
which students has full time teacher teaching  
each class separately.

company wants to  
severely reduce the burden of full time teacher in the institution  
(class)

so that it can be done by  
different form small groups or groups  
and each group is to take  
full unit of subject with their own

Output:-  
Student with highest marks: Anita - 470 with subject  
Student with marks above 400:  
Rahul - 456  
Karan - 420  
Anita - 470

£ £ £  
£ 300/-

£ 300

£ 600  
£ 200

```

highest - scorer = students [0]
for student in students :
    if student [1] > highest - scorer [1] :
        highest - scorer = student
print ("In Student with highest marks: {highest - scorer [0]}",
      (highest - scorer [1] ? marks))
print ("In students who scored above 400 marks:")
found - above - 400 = false
for student [ ] in students :
    if student [1] > 400 :
        print ("{student [0]} ({student [1]} ? marks)")
        found - above - 400 = True
if not found - above - 400 :
    print ("No Students scored above 400 marks")

```

else :

```
print ("No Student data entered")
```

### Result :-

The program successfully identifies the student with the highest marks and lists students scoring above 400 from a collection of student - mark tuples.

[[{"id": 1, "country": "India", "capital": "New Delhi"}, {"id": 2, "country": "USA", "capital": "Washington D.C."}, {"id": 3, "country": "Germany", "capital": "Berlin"}, {"id": 4, "country": "France", "capital": "Paris"}, {"id": 5, "country": "Japan", "capital": "Tokyo"}]]

((("Enter a country: ")))  
((("Enter its capital: ")))  
((("Enter country to search capital: ")))  
((("Capital of ")))  
((("is ")))  
((("Search result: ")))  
((("Country: ")))  
((("Capital: ")))

### 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

# 13/8/25 (c) Country - Capital Finder (Dictionary)

Aim:- To store 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 loop up.
5. Display all country - capital pairs sorted by country name.
6. End the program.

## Program:-

python

```
#Country -capital finder (Dictionary)
countries = { "India": "New Delhi", "France": "Paris", "Japan": "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 capital : ")
if search_country in countries:
    print ("capital of ", search_country, "is", countries [search_country])

else:
    print ("country not found :")
print ("In All country - Capital pairs (sorted) :")
for c in sorted (countries . keys ()):
    print (c, ":", countries [c])
```

Output

Result:-

The program to add, search and display country - capital pairs using dictionary was successfully executed.

VEL TECH - CSE	
X NO.	4
PERFORMANCE (5)	5
ESULT AND ANALYSIS (5)	5
IVA VOCE (5)	5
EARD (5)	5
TA	5