

Practical - 5

Aim:

Create, access, and modify the dictionary data structure.

Q1: Create and print a dictionary that contains keys a,b,c,d with their values 1,2,3 and 4 respectively using curly bracket syntax and ‘dict’ in built function.

Code:

```
d = {'a':1, 'b':2, 'c':3, 'd':4}
print(d)

d = dict([('a', 1), ('b', 2), ('c', 3), ('d', 4)])
print(d)
```

Output:

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

Q2: Using above created dictionary perform following operations

- Write a code to print out the value of a, d, and c.
- Calculate the sum of the value of a,b,c,d and print it.
- Add a new key, value pair (e,5) to the dictionary and print dictionary.

Code:

```
print(f"""
Values of a, b, c:
a: {d['a']}
b: {d['b']}
c: {d['c']}
""")
```

```
sum = 0
for i in d:
    sum += d[i]
print(f"Sum: {sum}\n")
```

```
d[ 'e' ] = 5
print(d)
```

Output:

Values of a, b, c:

a: 1

b: 2

c: 3

Sum: 15

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
```

Q3: Filter the dictionary by removing all items with a value greater than 2.

- d={"a":1, "b":2, "c":3, "d":4, "e":5}

Code:

```
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

```
l = []
for i in d:
    if d[i] > 2:
        l.append(i)
print(l)
```

```
for i in l:
    d.pop(i)
```

```
print(d)
```

Output:

```
['c', 'd']
{'a': 1, 'b': 2}
```

Q4: Print the names which contain the character 'a' from the dictionary containing 2 lists of male and female students given below.

- {"male": ["Tom", "Charlie", "Harry", "Frank"], "female": ["Sarah", "Huda", "Samantha", "Emily", "Elizabeth"] }

Code:

```
names = {"male": ["Tom", "Charlie", "Harry", "Frank"],  
"female": ["Sarah", "Huda", "Samantha", "Emily", "Elizabeth"] }  
  
a_names = []  
for val in names.values():  
    for i in val:  
        if 'a' in i:  
            a_names.append(i)  
  
print(a_names)
```

Output:

```
['Charlie', 'Harry', 'Frank', 'Sarah', 'Huda', 'Samantha',  
'Elizabeth']
```

Q5: You have 4 films in the dictionary with the age and number of seats available as indicated below. Write a programme to ask for a film and check for the person that he is eligible to watch movie, also check ticket availability and movie availability in the cinema.

- "War": [3,5],
- "Bourne": [18,5],
- "Gully boy": [15,5],
- "Uri": [12, 5]

Code:

```
movies = {"War": [3,5], "Bourne": [18,5], "Gully boy": [15,5],  
"Uri": [12, 5]}  
print("Given Movies: ", end="")  
for i in movies:  
    print(i, end=" ")  
  
user_query = input("\nEnter Movie You Want to Watch: ")  
  
if movies.get(user_query):  
    user_age = int(input("Enter Your Age: "))  
    if movies[user_query][0] <= user_age:  
        print("Person is Eligible to watch Movie.")  
        user_ticket_qunty = int(input("Enter How Many Ticket You Want:  
"))
```

```
if movies[user_query][1] >= user_ticket_qunty:  
    print("Tickets are available")  
else:  
    print("Tickets are not available")  
else:  
    print("Person is not Eligible to watch Movie.")  
else:  
    print("Movie is not in Cinema")
```

Output:

Given Movies: War Bourne Gully boy Uri

Enter Movie You Want to Watch: War

Enter Your Age: 18

Person is Eligible to watch Movie.

Enter How Many Ticket You Want: 5

Tickets are available

Practice Exercise:

1. Create a program for a student grading system using dictionary:

- Create a dictionary with student names as keys and a list of [attendance_percentage, marks] as values
- Add new student entries
- Remove students with attendance below 75%
- Print names of students who scored above 80 marks
- Display the dictionary after each operation

Code:

```

students_data = {
    "Alice Johnson": [92.5, 88],
    "Mark Smith": [71.0, 72],
    "Priya Sharma": [98.2, 95],
    "Liam Brown": [73.0, 81],
    "Sofia Garcia": [91.4, 89],
    "Honey Shah": [81.2, 98],
    "Vatsal Patel": [90, 95]
}

print("Original Dict:", students_data)

# Adding New Students

students_data["Sanjana Patel"] = [80.2, 89]

print("\nStudents Data after Adding new Student:", students_data)

# Removing Student with attendance below 75%
students_attendance_below_75 = []

for i in students_data:
    if students_data[i][0] < 75:
        students_attendance_below_75.append(i)

for i in students_attendance_below_75:
    students_data.pop(i)

print("\nStudents Data after Student attendance below 75%:", students_data)

# Displaying Student With More Score Then 80
print("\nStudents Who Have Scored Above 80: ", end="")
for i in students_data:
    if students_data[i][1] >= 80:
        print(i, end=", ")

```

Output:

```
Original Dict: {'Alice Johnson': [92.5, 88], 'Mark Smith': [71.0, 72], 'Priya Sharma': [98.2, 95], 'Liam Brown': [73.0, 81], 'Sofia Garcia': [91.4, 89], 'Honey Shah': [81.2, 98], 'Vatsal Patel': [90, 95]}
```

```
Students Data after Adding new Student: {'Alice Johnson': [92.5, 88], 'Mark Smith': [71.0, 72], 'Priya Sharma': [98.2, 95], 'Liam Brown': [73.0, 81], 'Sofia Garcia': [91.4, 89], 'Honey Shah': [81.2, 98], 'Vatsal Patel': [90, 95], 'Sanjana Patel': [80.2, 89]}
```

```
Students Data after Student attendance below 75%: {'Alice Johnson': [92.5, 88], 'Priya Sharma': [98.2, 95], 'Sofia Garcia': [91.4, 89], 'Honey Shah': [81.2, 98], 'Vatsal Patel': [90, 95], 'Sanjana Patel': [80.2, 89]}
```

Students Who Have Scored Above 80: Alice Johnson, Priya Sharma, Sofia Garcia, Honey Shah, Vatsal Patel, Sanjana Patel,

2. Write a program to manage a library catalog:

- books = { 'Python': ['John Smith', 5, 'Programming'], 'Digital Logic': ['Mike Ross', 3, 'Electronics'], 'Data Science': ['Sarah Connor', 0, 'Analytics'] }

Where values represent [author, copies_available, category]

- Add new books
- Update number of copies
- Print all books of specific category
- Remove books with 0 copies

Code:

```
books = { 'Python': ['John Smith', 5, 'Programming'],
          'Digital Logic': ['Mike Ross', 3, 'Electronics'],
          'Data Science': ['Sarah Connor', 0, 'Analytics'],
        }
```

```
print("Original Dict: ", books)
```

```

# Adding Books
books['Machine Learning'] = ['Anna Bell', 2, 'AI']
books['Web Development'] = ['Tom Hanks', 4, 'Programming']

print("\nDict After Adding New Books:", books)

# Update Number of Copies

books["Data Science"][1] = 2

print("\nDict After Updating Number of Copies:", books)

print("\nBooks Of Programming Catagory: ", end=' ')
for i in books:
    if books[i][2] == 'Programming':
        print(i, end=', ')

```

Output:

Original Dict: {'Python': ['John Smith', 5, 'Programming'], 'Digital Logic': ['Mike Ross', 3, 'Electronics'], 'Data Science': ['Sarah Connor', 0, 'Analytics']}

Dict After Adding New Books: {'Python': ['John Smith', 5, 'Programming'], 'Digital Logic': ['Mike Ross', 3, 'Electronics'], 'Data Science': ['Sarah Connor', 0, 'Analytics'], 'Machine Learning': ['Anna Bell', 2, 'AI'], 'Web Development': ['Tom Hanks', 4, 'Programming']}

Dict After Updating Number of Copies: {'Python': ['John Smith', 5, 'Programming'], 'Digital Logic': ['Mike Ross', 3, 'Electronics'], 'Data Science': ['Sarah Connor', 2, 'Analytics'], 'Machine Learning': ['Anna Bell', 2, 'AI'], 'Web Development': ['Tom Hanks', 4, 'Programming']}

Books Of Programming Catagory: Python, Web Development,

3. Create a program for a simple shopping cart:

- `cart = { 'apple': [50, 2], # [price_per_unit, quantity] 'mango': [80, 1], 'banana': [40, 3] }`
- Calculate total bill
- Update quantities
- Remove items with quantity 0
- Add new items
- Display cart status after each operation

Code:

```

def calculate_total_bill(cart):
    total_bill = 0
    for i in cart:
        total_bill += cart[i][0] * cart[i][1]
    return total_bill

# [price_per_unit, quantity]
cart = { 'apple': [50, 2], 'mango': [80, 1], 'banana': [40, 3] }
print("Original Cart:", cart)

# Total Bill
total_bill = 0
for i in cart:
    total_bill += cart[i][0] * cart[i][1]
print("Total Bill:", total_bill)

# Updating Quantity
cart['mango'][1] = 4
cart["banana"][1] = 0
print("\nCart After Updating Quantity:", cart)
print("Total Bill:", calculate_total_bill(cart))

# Removing Item with 0 quantity
quantity_0 = []
for i in cart:
    if cart[i][1] == 0:
        quantity_0.append(i)

for i in quantity_0:
    cart.pop(i)

print("\nCart After Removing 0 Quantity Item:", cart)
print("Total Bill:", calculate_total_bill(cart))

# Adding New Item
cart['chocolate'] = [100, 2]

print("\nCart After Adding New Item:", cart)

```

```
print("Total Bill:", calculate_total_bill(cart))
```

Output:

```
Original Cart: {'apple': [50, 2], 'mango': [80, 1], 'banana': [40, 3]}
```

```
Total Bill: 300
```

```
Cart After Updating Quantity: {'apple': [50, 2], 'mango': [80, 4], 'banana': [40, 0]}
```

```
Total Bill: 420
```

```
Cart After Removing 0 Quantity Item: {'apple': [50, 2], 'mango': [80, 4]}
```

```
Total Bill: 420
```

```
Cart After Adding New Item: {'apple': [50, 2], 'mango': [80, 4], 'chocolate': [100, 2]}
```

```
Total Bill: 620
```

4. Write a program that manages a contact directory: Create dictionary with name as key and [phone, email, city] as value

- Find all contacts from a specific city
- Update contact details
- Remove contacts not containing '@' in email
- Print contacts whose names start with given character

Code:

```

contacts = {
    "Alice": ["123-456-7890", "alice@example.com", "New York"],
    "Bob": ["987-654-3210", "bob@example.com", "Los Angeles"],
    "Charlie": ["555-555-5555", "charlie@example.com", "Chicago"],
    "Vatsal": ["111-222-3333", "vatsal@example.com", "Himmatnagar"],
    "Honey": ["444-444-4444", "honey@example.com", "Himmatnagar"]
}

print("Original Contacts: ", contacts)

# Find all contacts from a specific city
print("\nAll Contacts From Himmatnagar: ", end=' ')
for i in contacts:
    if contacts[i][2] == "Himmatnagar":
        print(i, end=', ')

# Update contact details
contacts['Charlie'][1] = 'Charlie Pvt LTD'
print("\n\nAfter Updating Contact Details:", contacts)

# Remove contacts not containing '@' in email
wrong_email = []
for i in contacts:
    if '@' not in contacts[i][1]:
        wrong_email.append(i)

for i in wrong_email:
    contacts.pop(i)

print("\nAfter Removing Wrong Email Details:", contacts)

```

Output:

```

Original Contacts: {'Alice': ['123-456-7890', 'alice@example.com', 'New York'],
                    'Bob': ['987-654-3210', 'bob@example.com', 'Los Angeles'],
                    'Charlie': ['555-555-5555', 'charlie@example.com', 'Chicago'],
                    'Vatsal': ['111-222-3333', 'vatsal@example.com', 'Himmatnagar'],
                    'Honey': ['444-444-4444', 'honey@example.com', 'Himmatnagar']}

```

```
'Himmatnagar'], 'Honey': ['444-444-4444', 'honey@example.com',  
'Himmatnagar']}
```

All Contacts From Himmatnagar: Vatsal, Honey,

```
After Updating Contact Details: {'Alice': ['123-456-7890',  
'alice@example.com', 'New York'], 'Bob': ['987-654-3210',  
'bob@example.com', 'Los Angeles'], 'Charlie': ['555-555-5555',  
'Charlie Pvt LTD', 'Chicago'], 'Vatsal': ['111-222-3333',  
'vatsal@example.com', 'Himmatnagar'], 'Honey': ['444-444-4444',  
'honey@example.com', 'Himmatnagar']}
```

```
After Removing Wrong Email Details: {'Alice': ['123-456-7890',  
'alice@example.com', 'New York'], 'Bob': ['987-654-3210',  
'bob@example.com', 'Los Angeles'], 'Vatsal': ['111-222-3333',  
'vatsal@example.com', 'Himmatnagar'], 'Honey': ['444-444-4444',  
'honey@example.com', 'Himmatnagar']}
```

5. Create a program for course registration system

- courses = { 'Python': ['Lab 1', 30, ['John', 'Mike']], 'Java': ['Lab 2', 25, ['Sarah', 'Tom']], 'Web Dev': ['Lab 3', 20, ['Alex']] }

Where values represent [lab_room, max_capacity, enrolled_students]

- Add new course
- Enroll student in course
- Remove course with less than 5 students
- Display available seats in each course

Code:

```

courses = { 'Python': ['Lab 1', 30, ['John', 'Mike']], 'Java': ['Lab 2', 25, ['Sarah', 'Tom']], 'Web Dev': ['Lab 3', 20, ['Alex']] }
print("Original Courses:", courses)

# Add New Courses
courses["Operating System"] = ['Lab 4', 20, []]
print("\nAfter Adding Course:", courses)

# Enrolling Students In course
courses['Operating System'][2].extend(['Vatsal', 'Aryan', 'Astha',
'Honey', 'Aditi', 'Sanjana'])
courses['Java'][2].extend(['Vatsal', 'Aryan', 'Astha', 'Honey'])
courses['Python'][2].extend(['Vatsal', 'Honey', 'Aditi', 'Sanjana'])
print("\nAfter Enrolling Student in Course:", courses)

# Removing Course with less than 5 student enrolled
less_enrolled_courses = []
for i in courses:
    if len(courses[i][2]) < 5:
        less_enrolled_courses.append(i)

for i in less_enrolled_courses:
    courses.pop(i)

print("\nAfter Removing Courses with less than 5 students: ",
courses)

for i in courses:
    print(f"\nCourse: {i} \nAvailable Seats: {courses[i][1] - len(courses[i][2])}")

```

Output:

```

Original Courses: {'Python': ['Lab 1', 30, ['John', 'Mike']], 'Java': ['Lab 2', 25, ['Sarah', 'Tom']], 'Web Dev': ['Lab 3', 20, ['Alex']]}

for i in courses:
    print(f"\nCourse: {i} \nAvailable Seats: {courses[i][1] - len(courses[i][2])}")

```

```
After Adding Course: {'Python': ['Lab 1', 30, ['John', 'Mike']],  
'Java': ['Lab 2', 25, ['Sarah', 'Tom']], 'Web Dev': ['Lab 3', 20,  
['Alex']], 'Operating System': ['Lab 4', 20, []]}
```

```
After Enrolling Student in Course: {'Python': ['Lab 1', 30, ['John',  
'Mike', 'Vatsal', 'Honey', 'Aditi', 'Sanjana']], 'Java': ['Lab 2',  
25, ['Sarah', 'Tom', 'Vatsal', 'Aryan', 'Astha', 'Honey']], 'Web  
Dev': ['Lab 3', 20, ['Alex']], 'Operating System': ['Lab 4', 20,  
['Vatsal', 'Aryan', 'Astha', 'Honey', 'Aditi', 'Sanjana']]}
```

```
After Removing Courses with less then 5 students: {'Python': ['Lab  
1', 30, ['John', 'Mike', 'Vatsal', 'Honey', 'Aditi', 'Sanjana']],  
'Java': ['Lab 2', 25, ['Sarah', 'Tom', 'Vatsal', 'Aryan', 'Astha',  
'Honey']], 'Operating System': ['Lab 4', 20, ['Vatsal', 'Aryan',  
'Astha', 'Honey', 'Aditi', 'Sanjana']]}
```

Course: Python

Available Seats: 24

Course: Java

Available Seats: 19

Course: Operating System

Available Seats: 14