**8 June 2024**

TOPICS:

**DICTIONARY**

**SETS**

1. You have a dictionary representing the inventory of a store with item names as keys and quantities as values. Add a new item to the inventory with a specific quantity.

```
inventory={'pen':20,'pencils':30,'notebooks':40}
inventory['eraser']=20

print(inventory)
```

```
{'pen': 20, 'pencils': 30, 'notebooks': 40, 'eraser': 20}
```

2. Given a dictionary of student names and their scores, find the score of a particular student.

```
student={'Raj':85,'kajal':88,'Deep':80}
print(student.get('kajal'))
```

```
88
```

3. You have a dictionary of country names as keys and their capitals as values. Update the capital of a specific country

```
country={'India':'New Delhi','France':'Paris','ShriLanka':'Kabul'}
new={'Shri Lanka':'Colombo'}
country.update(new)
print(country)
```

```
{'India': 'New Delhi', 'France': 'Paris', 'ShriLanka': 'Kabul', 'Shri Lanka': 'Colombo'}
```

4. Given a dictionary of product prices with product names as keys, remove a product from the dictionary

```
stationary_items={'pen':'20 Rs','pencils':'5 Rs','Notebook':'50 Rs'}
stationary_items.pop('pencils')
print(stationary_items)
```

```
{'pen': '20 Rs', 'Notebook': '50 Rs'}
```

5.You have two dictionaries representing two different departments with employee names as keys and their salaries as values. Merge these two dictionaries into one.

```
department_cse={'Raj':60000,'Jay':65000,'Nikhil':68000}
department_it={'Diviya':63000,'Raha':59000,'kajal':69000}
merge={**department_cse,**department_it}
print(merge)
```

```
{'Raj': 60000, 'Jay': 65000, 'Nikhil': 68000, 'Diviya': 63000, 'Raha': 59000, 'kajal': 69000}
```

6. Given a dictionary of city names and their populations, check if a particular city is in the dictionary.

```
cities={"New York":8765431,"San Antonio":1543287,"Houston":2387652}
cities_name="San Antonio"
if cities_name in cities:
  population=cities[cities_name]
  print(population)
else:
  print("no")
```

```
1543287
```

7. You have a dictionary of book titles as keys and their authors as values. Add multiple new books to the dictionary at once.

```
books={"To kill a Mockingbird":'Harper Lee',"The Great Gatsby":"F.Scott Fitzgerald"}
new_books ={"Pride and Prejudice":'Jane Austen',"The Lord of Rings":'J.R.R Tolklien'}
books.update(new_books)
books
```

```
{'To kill a Mockingbird': 'Harper Lee',
 'The Great Gatsby': 'F.Scott Fitzgerald',
 'Pride and Prejudice': 'Jane Austen',
 'The Lord of Rings': 'J.R.R Tolklien'}
```

8. Given a dictionary of students' names and their grades, find the student with the highest grade.

```
student={'Raj':85,'kajal':88,'Deep':80}
highest_grade=max(student)
print(highest_grade)
```

```
kajal
```

9.You have a dictionary representing a phone book with names as keys and phone numbers as values. Remove a contact from the phone book

```
student={'Raj':9876543212,'kajal':9867343488,'Deep':9801234560}
student.pop('kajal')
print(student)
```

```
{'Raj': 9876543212, 'Deep': 9801234560}
```

10. You have a dictionary of employee IDs as keys and their names as values. Update the name of an employee given their ID

```
employee={1:"RAJ", 2:"DIVYA",3:"DIYA"}
employee_id=2
new_name="MANAV"
if employee_id in employee:
  employee[employee_id]=new_name
  print(f"updated value {employee_id}'s name to {new_name}")
else:
  print(f"employee id {employee_id} not found")
print(employee)
```

```
updated value 2's name to MANAV
{1: 'RAJ', 2: 'MANAV', 3: 'DIYA'}
```

**SETS**

1. You have two sets representing students enrolled in Math and Science classes. Find the set of students who are enrolled in both classes.

```
student_maths={"Ram","Jay","Dev","Ishan"}
student_science={"Vijay","Raman","Deep","Manju"}
student_maths.union(student_science)
```

```
{'Deep', 'Dev', 'Ishan', 'Jay', 'Manju', 'Ram', 'Raman', 'Vijay'}
```

2. Given a set of unique employee IDs, add a new employee ID to the set

```
from ast import Add
#unique employee id also add new id to it
unique_id={345,234,566,677}
print(unique_id)
unique_id.add(456)
print(unique_id)
```

3. You are managing two different projects and have two sets representing the team members for each project. Find the set of team members who are working on either project or both projects.

```
project1={"Ram","Jay","Dev","Ishan"}
project2={"Vijay","Raman","Deep","Manju","Dev","Ishan"}
team_work_on_either_project=project1.union(project2)
print(team_work_on_either_project)
```

4. Given two sets of product IDs, one representing products in stock and the other representing products that have been sold, find the set of products that are still in stock.

```
product_present_stock={"product A","product B","product D"}
product_sold={"product C", "product D"}
update=product_present_stock.difference(product_sold)
print(update)
```

6. You have a set of registered conference attendees. Check if a particular person is registered.

```
regested={"Ram","Jay","Dev","Ishan"}
to_check_person={"Dev"}
to_check_person.issubset(regested)
```

7. Given two sets of book titles, one representing books available in the library and the other representing books borrowed by students, find the set of books that are not currently available in the library.

```
books_in_library={"The Marchant","On friendship","The Great Indian's","The World"}
books_borrowed={"The Great Indian's","The World"}

not_present_in_library=books_in_library.intersection(books_borrowed)
print(not_present_in_library)
```

8. You have a set of unique tags assigned to a blog post. Add multiple new tags to the set at once.

```
available_tags={"Python","Artifical Intelligence","Data Science"}
new_tags={"Data Anylsis","Data Structure"}
available_tags.update(new_tags)
print(available_tags)
```

9. Given a set of employee skills and another set of skills required for a new project, find the set of skills that need to be acquired or improved.

```
empolyee_skills={"Team Management","Communication","Problem solving"}
new_skills={"leadership","Adaptability"}
update=empolyee_skills.union(new_skills)
print(update)
```

10. You have two sets of favorite movies from two different friends. Find the set of movies that are favorite to at least one of them but not both.

```
favourite_movie1={"Lost","Khmoshi","Gaslight","1920"}
favourite_movie2={"Pari","1920","Dybbuk"}
update=favourite_movie1.intersection(favourite_movie2)
print(update)
```

11. Given two sets representing the collection of books in two different libraries, find the set of books that are available in both libraries.

```
books_in_library1={"Python","Artifical Intelligence","Data Science"}
books_in_library2={"Digital Marketing","Graphic Desigining","Data Structure and Algorithms"}
total_available=books_in_library1|books_in_library2
print(total_available)
```