

✓ Data Science - Summer 2025

Course: Data Science | **Instructor:** Dr. Adil Khan | **Submitted by:** Umair Saad - 023-21-0182

Python (Lists, Tuples, Sets, and Dictionaries) - Assignment - 02

TASK-1

```
myCities = ["Ghotki", "Sukkur", "Mirpurkhas", "Karachi", "Lahore"]
print("Five Cities: ", myCities)
# Add a city
myCities.append("Digri")
print("After add new city: ", myCities)
myCities.remove("Lahore")
print("After remove city: ", myCities)
```

⇒ Five Cities: ['Ghotki', 'Sukkur', 'Mirpurkhas', 'Karachi', 'Lahore']
 After add new city: ['Ghotki', 'Sukkur', 'Mirpurkhas', 'Karachi', 'Lahore', 'Digri']
 After remove city: ['Ghotki', 'Sukkur', 'Mirpurkhas', 'Karachi', 'Digri']

TASK-2

```
evenNumbers = (2, 4, 6, 8, 10)
print("Even Numbers Tuple: ", evenNumbers)
```

#modifying

```
try:
    evenNumbers[2] = 12
except TypeError as e:
    print("Error occurred while modifying tuple:", e)
    print("Explanation: Tuples are immutable, so their elements cannot be changed once de
```

Access 3rd element

```
thirdElement = evenNumbers[2]
print("The 3rd element is:", thirdElement)
```

⇒ Even Numbers Tuple: (2, 4, 6, 8, 10)
 Error occurred while modifying tuple: 'tuple' object does not support item assignment
 Explanation: Tuples are immutable, so their elements cannot be changed once defined.
 The 3rd element is: 6

TASK-3

```
favoriteFoods = {"Biryani", "Pizza", "Burger", "Pasta", "Biryani"}
print("Favorite Foods Set:", favoriteFoods)
```

```
print(f'Favorite Foods Set: {favorite_foods}, favorite_foods,
```

```
# Explanation:
```

```
print("Explanation: Repeated items automatically removed.")
```

```
Favorite Foods Set: {'Pasta', 'Pizza', 'Burger', 'Biryani'}
Explanation: Repeated items automatically removed.
```

```
# TASK-4 Dictionary Basics (Umair Saad --- 023-21-0182)
```

```
student = {"firstName": "Ali", "lastName": "Khan", "age": 20, "city": "Sukkur", "grade": "A"}
print("Student Dictionary:", student)
```

```
# all keys
```

```
print("Keys:", student.keys())
```

```
# all values
```

```
print("Values:", student.values())
```

```
# Print value of 'grade'
```

```
print("Grade:", student["grade"])
```

```
Student Dictionary: {'firstName': 'Ali', 'lastName': 'Khan', 'age': 20, 'city': 'Sukk
Keys: dict_keys(['firstName', 'lastName', 'age', 'city', 'grade'])
Values: dict_values(['Ali', 'Khan', 20, 'Sukkur', 'A'])
Grade: A
```

Data Science - Summer 2025 (Umair Saad --- 023-21-0182)

```
# TASK-5
```

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print("List:", numbers)
```

```
print("First 5 numbers:", numbers[:5])
```

```
print("Last 3 numbers:", numbers[-3:])
```

```
# Replace (index 1) with 99
```

```
numbers[1] = 99
```

```
print("After replacing 2nd with 99:", numbers)
```

```
List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
First 5 numbers: [1, 2, 3, 4, 5]
Last 3 numbers: [8, 9, 10]
After replacing 2nd with 99: [1, 99, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# TASK-6
```

```
insanInfo = ("Ali", 20, "Karachi")
print("OG Tuple:", insanInfo)

# Unpack
name, age, city = insanInfo

print("Name:", name)
print("Age:", age)
print("City:", city)
```

```
OG Tuple: ('Ali', 20, 'Karachi')
Name: Ali
Age: 20
City: Karachi
```

```
# TASK-7
```

```
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
print("Set A:", A)
print("Set B:", B)
```

```
# union - A and B
unionAB = A.union(B)
print("Union (A ∪ B):", unionAB)
```

```
# intersection - A and B
intersectionAB = A.intersection(B)
print("Intersection (A ∩ B):", intersectionAB)
```

```
# difference (A - B)
differenceAB = A.difference(B)
print("Difference (A - B):", differenceAB)
```

```
Set A: {1, 2, 3, 4}
Set B: {3, 4, 5, 6}
Union (A ∪ B): {1, 2, 3, 4, 5, 6}
Intersection (A ∩ B): {3, 4}
Difference (A - B): {1, 2}
```

```
# TASK-8
```

```
student = {"name": "Ali", "age": 20, "grade": "A"}
print("Before Updated", student)
# new key> email
student["email"] = "ali@example.com"
```

```
# Update the grade
... ..
```

```
student["grade"] = "A+"
```

```
# Remove age
```

```
student.pop("age")
```

```
print("Updated Student Dictionary:", student)
```

```
Before Updated {'name': 'Ali', 'age': 20, 'grade': 'A'}
```

```
Updated Student Dictionary: {'name': 'Ali', 'grade': 'A+', 'email': 'ali@example.com'}
```

```
# TASK-9
```

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
```

```
even_numbers = []
```

```
for num in nums:
```

```
    if num % 2 == 0:
```

```
        even_numbers.append(num)
```

```
print("Even Numbers:", even_numbers)
```

```
Even Numbers: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
```

```
# TASK-10
```

```
students = {
```

```
    "student1": {"name": "Sara", "age": 19, "marks": 85},
```

```
    "student2": {"name": "Zain", "age": 21, "marks": 92}
```

```
}
```

```
for key in students:
```

```
    name = students[key]["name"]
```

```
    marks = students[key]["marks"]
```

```
    print(f"{name} scored {marks} marks.")
```

```
Sara scored 85 marks.
```

```
Zain scored 92 marks.
```

```
# TASK-11
```

```
products = [
```

```
    {"item": "Book", "price": 300},
```

```
    {"item": "Pen", "price": 50},
```

```
    {"item": "Bag", "price": 1500}
```

```
]
```

```
for product in products:
```

```
    print(f"{product['item']} - Rs.{product['price']}")
```

```
Book - Rs.300
```

```
Pen - Rs.50
```

```
Pen - Rs.50  
Bag - Rs.1500
```

TASK-12

```
cricket = {"Ali", "Sara", "Zain"}  
football = {"Zain", "Fatima", "Ali"}  
  
both = cricket.intersection(football)  
print("Play both:", both)  
  
only_cricket = cricket.difference(football)  
print("Only cricket:", only_cricket)  
  
only_one_sport = cricket.symmetric_difference(football)  
print("Only one sport:", only_one_sport)
```

TASK-13

```
squares = [x**2 for x in range(1, 6)]  
print("Squares from 1 to 5:", squares)  
  
Squares from 1 to 5: [1, 4, 9, 16, 25]
```

TASK-14

```
fruits = ["apple", "banana", "apple", "orange", "banana", "apple"]  
count_apple = fruits.count("apple")  
print("Apple", count_apple, "times.")  
  
count = fruits.count("orange")  
print("Orange", count, "times.")  
  
count = fruits.count("banana")  
print("Banana", count, "times.")  
  
Apple 3 times.  
Orange 1 times.  
Banana 2 times.
```

TASK-15

```
todo = {}  
  
while True:  
    print("\n1. Add Task\n2. View Tasks\n3. Mark as Completed\n4. Remove Completed Tasks\  
    choice = input("Choose an option: ")
```

```

if choice == "1":
    task = input("Enter task name: ")
    todo[task] = "Incomplete"
elif choice == "2":
    for task, status in todo.items():
        print(f"{task}: {status}")
elif choice == "3":
    task = input("Enter task to mark as completed: ")
    if task in todo:
        todo[task] = "Completed"
    else:
        print("Task not found.")
elif choice == "4":
    completed_tasks = [task for task, status in todo.items() if status == "Completed"]
    for task in completed_tasks:
        del todo[task]
    print("Completed tasks removed.")
elif choice == "5":
    break
else:
    print("Invalid choice.")

```

```

1. Add Task
2. View Tasks
3. Mark as Completed
4. Remove Completed Tasks
5. Exit
Choose an option: 5

```

TASK-16

```
contacts = {}
```

```

while True:
    print("\n1. Add Contact\n2. View Contacts\n3. Search Contact\n4. Exit")
    choice = input("Choose an option: ")

    if choice == "1":
        name = input("Enter name: ")
        phone = input("Enter phone number: ")
        contacts[name] = phone
    elif choice == "2":
        for name, phone in contacts.items():
            print(f"{name}: {phone}")
    elif choice == "3":
        search_name = input("Enter name to search: ")
        if search_name in contacts:
            print(f"{search_name}'s number is {contacts[search_name]}")
        else:

```

```
        else:
            print("Contact not found.")
    elif choice == "4":
        break
    else:
        print("Invalid option.")
```

```
1. Add Contact
2. View Contacts
3. Search Contact
4. Exit
Choose an option: 4
```

Data Science - Summer 2025

Course: Data Science | **Instructor:** Dr. Adil Khan | **Submitted by:** Umair Saad - 023-21-0182

Python (Lists, Tuples, Sets, and Dictionaries) - Assignment - 02