**Lab#7**

**Q#1:** Write a program that asks the user for student’s name and obtained marks for 10 students and then stores them in a dictionary.

**Input:**

Student\_name = (input(‘Enter Student Name:’))

Obtained\_marks = {

‘Jawaria’ : 90,

‘Haiqa’ : 80,

‘Zainab’ : 70,

‘Firza’ : 60,

‘Bushra’ : 50,

‘Hava’ : 89,

‘BiBi’ : 87,

‘Fatima’ : 56,

‘Sadia’ : 98,

‘Laiqa’ : 99

}

Print(obtained\_marks[student\_name])

Output:

Enter Student Name:Jawaria

90

[Program finished]

**Q#2:** Check for a duplicate key before saving a new item in the dictionary. In case anitem is already available in the students\_dict, then the program should warn the user.

**Input:**

Students\_dict = {

101: ‘Jawaria’,

105: ‘Zainab’,

108: ‘Hava’

}

Def add\_student(id, name):

If id in students\_dict:

Print(“Warning: Student with ID”, id, “already exists.”)

Else:

Students\_dict[id] = name

Print(“Student added successfully.”)

Add\_student(101, ‘Ali’)

Add\_student(103, ‘John’)

Print(students\_dict)

**Output:**

Warning: Student with ID 101 already exists.

Student added successfully.

{101: ‘Jawaria’, 105: ‘Zainab’, 108: ‘Hava’, 103: ‘John’}

[Program finished]

Q#3: To use a unique key for each record. This way, every student will be assigned a

Unique number and two students with same name can be added to the students\_dict.

**Input:**

Students\_dict = {}

Next\_student\_id = 1

Def add\_student(name):

Global next\_student\_id

Students\_dict[next\_student\_id] = name

Print(“Student”, name, “added successfully with ID”, next\_student\_id)

Next\_student\_id += 1 # Increment the student ID counter for the next student

Add\_student(‘Jawaria’)

Add\_student(‘Hava’)

Add\_student(‘Sadia’)

Print(students\_dict)

**Output:**

Student Jawaria added successfully with ID 1

Student Hava added successfully with ID 2

Student Sadia added successfully with ID 3

{1: ‘Jawaria’, 2: ‘Hava’, 3: ‘Sadia’}

[Program finished]

**Q#4:** Consider the following dictionary that contains the items id and item’s price in aShop: Item\_dict={‘item\_1’: 45.50, ‘item\_2’:35, ‘item\_3’: 41.30, ‘item\_4’:55, ‘item\_5’: 24} Write a program that finds the item with (1) highest price, and (2) smallest price.

**Input:**

Item\_dict = {‘item\_1’: 45.50, ‘item\_2’: 35, ‘item\_3’: 41.30, ‘item\_4’: 55, ‘item\_5’: 24}

# Initialize variables to store the item with the highest and lowest prices

Max\_price\_item = None

Min\_price\_item = None

# Iterate through the dictionary to find the item with the highest and lowest prices

For item, price in Item\_dict.items():

If max\_price\_item is None or price > Item\_dict[max\_price\_item]:

Max\_price\_item = item

If min\_price\_item is None or price < Item\_dict[min\_price\_item]:

Min\_price\_item = item

# Print the results

Print(“Item with highest price:”, max\_price\_item, “($”, Item\_dict[max\_price\_item], “)”)

Print(“Item with smallest price:”, min\_price\_item, “($”, Item\_dict[min\_price\_item], “)”)

**Q#5:** Initialize the following list and dictionaries in the program:

Car\_ids = [1,2,3,4,5]

Car\_names = {1:”Suzuki Cultus”,2:”Suzuki Alto”,3:”Toyota Corolla”, 4:”Honda City”, 5:”Honda Civic”}

Car\_models = {1:2020,2:2021,3:2020,4:2021,5:2020}

Car\_rentals = {1:2000,2:1500,3:3000,4:2500,5:3500}

Car\_fuel = {1:17,2:19,3:13,4:14,5:12}

Car\_issued = {1:0,2:0,3:0,4:0,5:0}

Now, write a program that implements a car-rental system. The program shows a menu similar to the Following: The user can then make a choice to rent a car. When the user selects a car id to rent, the program asks for Customer name and CNIC and updates the status of the car. When a car is rented, the program updates the Menu and prints it again. The program exits when the user selects ‘q’ as input.

**Input:**

Car\_ids = [1, 2, 3, 4, 5]

Car\_names = {1: “Suzuki Cultus”, 2: “Suzuki Alto”, 3: “Toyota Corolla”, 4: “Honda City”, 5: “Honda Civic”}

Car\_models = {1: 2020, 2: 2021, 3: 2020, 4: 2021, 5: 2020}

Car\_rentals = {1: 2000, 2: 1500, 3: 3000, 4: 2500, 5: 3500}

Car\_fuel = {1: 17, 2: 19, 3: 13, 4: 14, 5: 12}

Car\_issued = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0}

Def rent\_car(car\_id):

If car\_id not in car\_ids:

Print(“Invalid car ID.”)

Return

If car\_issued[car\_id] == 1:

Print(“Car is already rented out.”)

Return

Customer\_name = input(“Enter customer name: “)

Customer\_cnic = input(“Enter customer CNIC: “)

Car\_issued[car\_id] = 1

Print(f”Car {car\_names[car\_id]} rented successfully to {customer\_name} (CNIC: {customer\_cnic}).”)

Def display\_menu():

Print(“\nCar Rental Menu:”)

Print(“ID Name Model Rental (per day)”)

For car\_id in car\_ids:

Print(f”{car\_id}: {car\_names[car\_id]:<16} {car\_models[car\_id]} ${car\_rentals[car\_id]}”)

# Main program

While True:

Display\_menu()

Choice = input(“\nEnter car ID to rent (or ‘q’ to quit): “)

If choice == ‘q’:

Break

Else:

Try:

Car\_id = int(choice)

Rent\_car(car\_id)

Except ValueError:

Print(“Invalid input. Please enter a valid car ID or ‘q’ to quit.”)

**Output:**

Car Rental Menu:

ID Name Model Rental (per day)

1: Suzuki Cultus 2020 $2000

2: Suzuki Alto 2021 $1500

3: Toyota Corolla 2020 $3000

4: Honda City 2021 $2500

5: Honda Civic 2020 $3500

Enter car ID to rent (or ‘q’ to quit):

**Q#6:** Write a program that inputs username and password from the user, and then matches With the stored username:password in a dictionary. The program welcomes the user if the username and Password are correct and regrets when wither username or password is incorrect.

**Input:**

# Dictionary containing username-password pairs

User\_credentials = {

‘user1’: ‘password1’,

‘user2’: ‘password2’,

‘user3’: ‘password3’

}

Def authenticate(username, password):

If username in user\_credentials and user\_credentials[username] == password:

Return True

Else:

Return False

Def main():

While True:

Username = input(“Enter username: “)

Password = input(“Enter password: “)

If authenticate(username, password):

Print(“Welcome,”, username)

Break

Else:

Print(“Incorrect username or password. Please try again.”)

If \_\_name\_\_ == “\_\_main\_\_”:

Main()

**Q#7:** Write a program that lets the user input a sentence and then finds the frequency of Alphabets in the entered sentence. The output should be similar to the following:

**Input:**

User\_credentials = {

‘user1’: ‘password1’,

‘user2’: ‘password2’,

‘user3’: ‘password3’

}

Def authenticate(username, password):

If username in user\_credentials and user\_credentials[username] == password:

Return True

Else:

Return False

Def main():

While True:

Username = input(“Enter username: “)

Password = input(“Enter password: “)

If authenticate(username, password):

Print(“Welcome,”, username)

Break

Else:

Print(“Incorrect username or password. Please try again.”)

If \_\_name\_\_ == “\_\_main\_\_”:

Main()

**Output:**

Enter username: Jawaria

Enter password: 1234

Incorrect username or password. Please try again.

**QUESTIONS:**

**Q # 1: Which is the correct form of declaration of dictionary?**

a) day = {1: 'm', 2: 't', 3: 'w'}

b) day = (1; 'm', 2; 't', 3; 'w')

c) day = [1: 'm', 2: 't', 3: 'w']

d) day = {1 'm', 2 't', 3 'w'}

**Ans.**

day = {1: 'm', 2: 't', 3: 'w'}

**Q # 2: Consider a dictionary dates is declared as:**

dates={2000:"15.04.2020", 2001:"26.03.2001", 2002:"17.08.2002"}

Write the code that changes the date items of the year 2002 to “20.05.2002”

**Ans**.

dates = {2000: "15.04.2020", 2001: "26.03.2001", 2002: "17.08.2002"}

dates[2002] = "20.05.2002"

**Q # 3: Write the output of following code:**

x = {1:10}

d = {2:20, 3:30, 4:40}

x.update(d)

print(x)

**Ans**.

{1: 10, 2: 20, 3: 30, 4: 40}

**Q # 4: Write the output of following code:**

d = {'x': 1, 'y': 2, 'z': 3}

a = d.pop('y')

print(a)

print(d)

**Ans**

{'x': 1, 'z': 3}