**CSE Week-6:**

**26. Write a Python program that performs the following operations on a**

**Dictionaries a) Create a dict**

my\_dict = {"a": 10, "b": 20, "c": 30, "d": 40, "e": 50}

**b) Print the index of the first element**

first\_key = next(iter(my\_dict))

first\_key

first\_index = list(my\_dict.keys()).index(first\_key)

first\_index

**Note:Alternative Key**

index\_of\_b = list(my\_dict.keys()).index("b")

index\_of\_b

**c)Remove the 3rd element in the dict**

third\_key = list(my\_dict.keys())[2]

third\_key

del my\_dict[third\_key]

my\_dict

27. Write a Python program that performs the following operations on a

Dictionaries.

**a)Print the first element to fourth element in the dict.**

first\_four\_items = list(my\_dict.items())[:4]

first\_four\_items

**b)print the first element from the rightmost index.**

last\_item = list(my\_dict.items())[-1]

last\_item

**c) reverse the element in the dict**

reversed\_dict = dict(reversed(my\_dict.items()))

reversed\_dict

d) Add the element in the dict.

my\_dict["f"] = 60

my\_dict

28. Write a Python program that demonstrates implicit type conversion.

In this program, add an integer to a float, and print the result.

int\_num = 10

float\_num = 5.5

result = int\_num + float\_num

result

type(result)

29. Write a Python program that demonstrates explicit type conversion by

converting **a) float to an integer**

float\_num = 9.7

int\_num = int(float\_num)

float\_num

int\_num

type(int\_num)

**b) integer to hex**

num = 255

hex\_value = hex(num)

hex\_value

type(hex\_value)

**c) integer to octal**

num = 255

oct\_value=oct(num)

oct\_value

type(oct\_value)

30. Convert a decimal number into binary,octal, and hexa decimal number

system.

num = 255

binary\_value = bin(num)

binary\_value

octal\_value = oct(num)

octal\_value

hexadecimal\_value = hex(num)

hexadecimal\_value

ASSESMENT: Student Grade Management System

Objective: Create a program that allows you to:

1. Store student names and their grades.

student\_grades = {}

def add\_student(name, grade):

student\_grades[name] = grade

print(f"Added: {name} with grade {grade}")

def get\_grade(name):

return student\_grades.get(name, "Student not found")

add\_student("Rahul", "A")

print("Grade",get\_grade("Rahul"))

2. Perform operations on lists (such as storing and displaying

students).

students = []

def add\_student(name):

students.append(name)

print(f"Added: {name}")

def display\_students():

if not students:

print("No students in the list.")

else:

print("\nStudent List:")

for index, name in enumerate(students, start=1):

print(f"{index}. {name}")

add\_student("Rakesh")

add\_student("Bunny")

display\_students()

3.Use tuples to represent immutable grade data for each student.

students = [("Sita", "A"), ("Radha", "B"), ("Gopika", "A")]

def display\_students():

print("\nStudent Grades:")

for name, grade in students:

print(f"{name}: {grade}")

def get\_grade(name):

for student in students:

if student[0] == name:

return f"{name}'s Grade: {student[1]}"

return "Student not found"

print(get\_grade("Sita"))

4. **Use a dictionary to store student names as keys and their grades**

**(as tuples) as values.**

student\_grades = {

"Kamalesh": ("A", "B", "A"),

"Srinivasa": ("B", "C", "B"),

"Charun": ("A", "A", "A")

}

def display\_students():

print("\nStudent Grades:")

for name, grades in student\_grades.items():

print(f"{name}: {grades}")

def get\_grades(name):

return student\_grades.get(name, "Student not found")

display\_students()

print("\nCharlie's Grades:", get\_grades("Charun"))

5**. Calculate the average grade for each student.**

student\_grades = {

"Anil": ("A", "B", "A"),

"kishore": ("B", "C", "B"),

"Satyam": ("A", "A", "A")

}

# Grade point mapping (A = 4, B = 3, C = 2, D = 1, F = 0)

grade\_points = {"A": 4, "B": 3, "C": 2, "D": 1, "F": 0}

# Function to calculate average grade

def calculate\_average(grades):

numeric\_grades = [grade\_points[grade] for grade in grades] # Convert letter grades to points

return sum(numeric\_grades) / len(numeric\_grades) # Compute average

# Function to display student averages

def display\_averages():

print("\nStudent Average Grades:")

for name, grades in student\_grades.items():

avg = calculate\_average(grades)

print(f"{name}: {avg:.2f}")

# Display all student averages

display\_averages()