

```
# (i) Write a Python program to swap 4 variables values
a, b, c, d = 2, 56, 78, 9

print("before swapping:")
print(f"a={a}, b={b}, c={c}, d={d}")
# before swapping:
# a=2, b=56, c=78, d=9

# swapping
a, b, c, d = d, c, b, a

# values after swapping
print("after swapping:")
print(f"a={a}, b={b}, c={c}, d={d}")
# after swapping:
# a=9, b=78, c=56, d=2


# (ii) Write a Python program to convert temperatures to and from celsius,

celsius = float(input("Enter temp in Celsius: "))

# conversion
fahrenheit = (celsius * 9 / 5) + 32

print(f"Temperature in Fahrenheit is: {fahrenheit}")
# Enter temp in Celsius: 60
# Temperature in Fahrenheit is: 140.0
```

```
# (i) Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of
sample_list = ['abc', 'xyz', 'aba', '1221']

count = 0

for s in sample_list:
    if len(s) >= 2 and s[0] == s[-1]:
        count += 1

print("Number of strings meeting the criteria:", count)
# Number of strings meeting the criteria: 2

# (ii) Write a list comprehension which, from a list, generates a lowercased version of each string that has length greater than five.

list = ['HelloWorld', 'Python', 'AI', 'DeepLearning']

result = []
for s in sample_list:
    if len(s) > 5:
        result.append(s.lower())

print("Lowercased strings with length > 5:", result)
# Lowercased strings with length > 5: []

# (iii) Write a Python program to print a specified list after removing the 0th, 4th and 5th elements

sample_list = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow', 'Teapink']

result = []
for i in range(len(sample_list)):
    if i not in [0, 4, 5]:
        result.append(sample_list[i])

print("List after removing 0th, 4th, and 5th elements:", result)
# List after removing 0th, 4th, and 5th elements: ['Green', 'White', 'Black', 'Teapink']
```

```

# (i) Use dir and help to learn about the functions you can call on dictionaries and implement it.
dict = {'Ahsan': 29, 'Mustafa': 24, 'Arham': 16}

copied_dict = dict.copy()
print("Copy of dictionary:", copied_dict)
# Copy of dictionary: {'Ahsan': 29, 'Mustafa': 24, 'Arham': 16}

new_dict = dict.fromkeys(['Arham', 'Mustafa', 'Ahsan'], 3.6) # Create a new dictionary with keys and a default value
print("From keys:", new_dict)
# From keys: {'Arham': 3.6, 'Mustafa': 3.6, 'Ahsan': 3.6}

value = dict.get('Rahim', 'Not Found') # Get the value for a key, with a default value if the key doesn't exist
print("Value for key 'b': ", value)
# Value for key 'b': Not Found

print("Items in dictionary:", dict.items())
# Items in dictionary: dict_items([('Ahsan', 29), ('Mustafa', 24), ('Arham', 16)])

print("Keys in dictionary:", dict.keys())
# Keys in dictionary: dict_keys(['Ahsan', 'Mustafa', 'Arham'])

pop_value = dict.pop('Suleman', 'Key not found')
print("After pop('c'):", dict)
# After pop('c'): {'Ahsan': 29, 'Mustafa': 24, 'Arham': 16}
print("Popped value:", pop_value)
# Popped value: Key not found

last_item = dict.popitem() #Remove and return the last key-value pair
print("After popitem:", dict)
# After popitem: {'Ahsan': 29, 'Mustafa': 24}
print("Last item:", last_item)
# Last item: ('Arham', 16)

# remaking the dictionary
dict = {'Kamran': 21, 'Abdullah': 19, 'Huzaifa': 21}

default_value = dict.setdefault('Zawad', 19)
print("After setdefault:", dict)
# After setdefault: {'Kamran': 21, 'Abdullah': 19, 'Huzaifa': 21, 'Zawad': 19}

dict.update({'Wajahat': 20, 'Zafar': 21})
print("After update:", dict)
# After update: {'Kamran': 21, 'Abdullah': 19, 'Huzaifa': 21, 'Zawad': 19, 'Wajahat': 20, 'Zafar': 21}

dict.clear()
print("After clear():", dict)
# After clear(): {}

# (ii) Write a Python script to concatenate following dictionaries to create a new one.
dic1 = {1: 10, 2: 20}
dic2 = {3: 30, 4: 40}
dic3 = {5: 50, 6: 60}

result = {**dic1, **dic2, **dic3}

print("Concatenated Dictionary:", result)
# Concatenated Dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

```

1. Identity Operators in Python:

```
x = 6
if (type(x) is int):
    print ("true")
else:
    print ("false")
```

Output: True

```
x = 7.2
if (type(x) is not int):
    print ("true")
else:
    print ("false")
```

Output: True

3. Membership Operator:

```
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9]
for item in list1:
    if item in list2:
        print("overlapping")
else:
    print("not overlapping")
```

Output: not overlapping

4. Floor Division and Exponentiation:

```
a = 5
a //= 3 # floor division
a **= 5 # exponentiation
print("floor divide=", a)
print("exponent=", a)
```

Output: floor divide= 1

Output: exponent= 1

5. Bitwise Operators:

```
a = 60 # 60 = 0011 1100
b = 13 # 13 = 0000 1101
c = 0
```

```
c = a & b # Bitwise AND (12 = 0000 1100)
print("Line 1", c)
```

```
c = a | b # Bitwise OR (61 = 0011 1101)
print("Line 2", c)
```

```
c = a ^ b # Bitwise XOR (49 = 0011 0001)
print("Line 3", c)
```

```
c = ~a # Bitwise NOT (-61 = 1100 0011)
print("Line 4", c)
```

```
c = a << 2 # Bitwise Left Shift (240 = 1111 0000)
print("Line 5", c)
```

```
c = a >> 2 # Bitwise Right Shift (15 = 0000 1111)
print("Line 6", c)
```

Output:

Line 1 12

```
# Line 2 61
# Line 3 49
# Line 4 -61
# Line 5 240
# Line 6 15
```

This program is a basic student management system that allows adding, viewing, and updating student records.

```
print("Welcome to the Student Management System!")

students = []

def add_student(name, age, grades):
    """Function to add a student record."""
    student = {
        'name': name,
        'age': age,
        'grades': grades
    }
    students.append(student)

def view_students():
    """Function to view all students."""
    if not students:
        print("No students in the system.")
    else:
        for student in students:
            print(f"Name: {student['name']}, Age: {student['age']}, Grades: {student['grades']}")

def update_student(name, age=None, grades=None):
    """Function to update a student's record."""
    for student in students:
        if student['name'] == name:
            if age:
                student['age'] = age
            if grades:
                student['grades'] = grades
            print(f"Updated record for {name}.")
            return
    print(f"Student {name} not found.")

def delete_student(name):
    """Function to delete a student by name."""
    for student in students:
        if student['name'] == name:
            students.remove(student)
            print(f"Student {name} has been deleted.")
            return
    print(f"Student {name} not found.")

while True:
    print("\n1. Add Student")
    print("2. View Students")
    print("3. Update Student")
    print("4. Delete Student")
    print("5. Exit")

    choice = input("Enter your choice (1/2/3/4/5): ")

    if choice == '1':
        name = input("Enter student's name: ")
        age = int(input("Enter student's age: "))
        grades = input("Enter student's grades (comma separated, Like Math:90, Science:85): ")

        if grades:
            grades_list = grades.split(",")
            grades = []
            for grade in grades_list:
                grades.append(tuple(grade.split(":")))
        else:
            grades = None

        grades = [(subject.strip(), int(grade.strip())) for subject, grade in grades]
        add_student(name, age, grades)
        print(f"Student {name} added successfully.")

    elif choice == '2':
        print("\n--- Student Records ---")
```

```
view_students()

elif choice == '3':
    name = input("Enter the student name to update: ")
    age = input("Enter new age: ")
    grades = input("Enter new grades: ")

    age = int(age) if age else None

    if grades:
        grades_list = grades.split(",")
        grades = []
        for grade in grades_list:
            grades.append(tuple(grade.split(":")))
    else:
        grades = None

    if grades:
        updated_grades = []
        for subject, grade in grades:
            updated_grades.append((subject.strip(), int(grade.strip())))
        grades = updated_grades
    else:
        grades = None

    update_student(name, age, grades)

elif choice == '4':
    name = input("Enter the student's name to delete: ")
    delete_student(name)

elif choice == '5':
    print("Exiting program...")
    break

else:
    print("Invalid choice, please try again.")
```