# Experiment No 7

# Student Record Keeper

**Aim –** To write a Python program to create, update, and manipulate a dictionary of student records, including their grades and attendance.

**Theory** : - The program manages a dictionary of student records, where each student's name is a key, and their corresponding value is another dictionary containing grades and attendance information. It allows the user to add new student records, update grades or attendance for existing students, and view all student records. This program demonstrates the use of nested dictionaries for efficient data organization and manipulation.

## Algorithm for the Program:

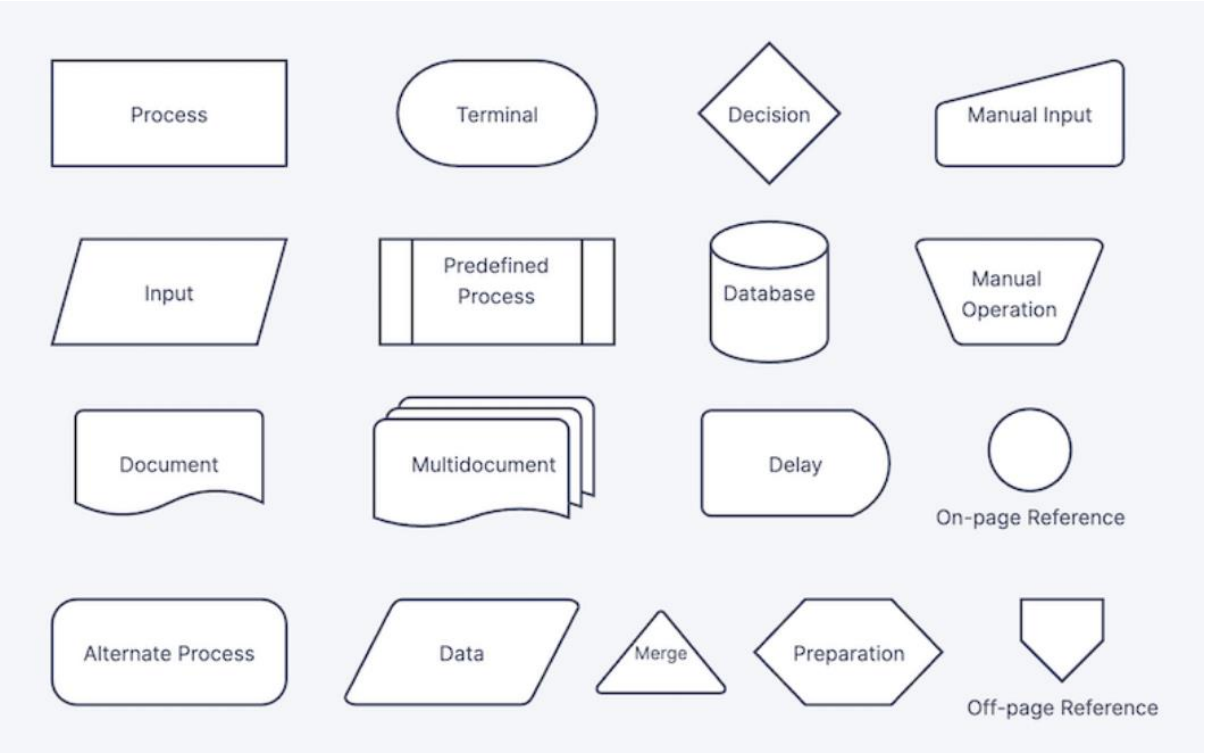
- • **Initialization**:
    - Create an empty dictionary `student` to store student records.
- **Add Student**:
    - Prompt the user for the student's name, grade, and attendance percentage.
    - Add the student record to the `students` dictionary with the name as the key and a nested dictionary (`{"grade": grade, "attendance": attendance}`) as the value.
    - Display a success message.
- **Update Student**:
    - Prompt the user for the name of the student whose record needs updating.
    - Check if the student's name exists in the dictionary:
        - If found, prompt the user for the new grade and attendance percentage.
        - Update the corresponding record in the `students` dictionary.
        - Display a success message.
    - If the student's name is not found, display an error message.
- **Display Students**:
    - Iterate through the `students` dictionary:
        - For each student, print their name, grade, and attendance percentage.
    - If the dictionary is empty, display a message indicating no records are available.
- **Interactive Menu**:
    - Display a menu with the following options:
        1. Add a student.
        2. Update a student record.
        3. Display all student records.
        4. Exit the program.
    - Prompt the user for their choice.
    - Perform the corresponding operation based on the choice:
        - Call `add_student()`, `update_student()`, or `display_students()` as needed.
        - Break the loop and terminate the program if the user selects "Exit".

- Display an error message for invalid choices.

- **Repeat**:
  - Continue displaying the menu and processing user input until the user chooses to exit.

## Flowchart

## Draw flowchart using following symbols.




**Program : -**

```
students = {}
def add_student():
    name = input("Enter student name: ")
    grade = input("Enter grade: ")
    attendance = int(input("Enter attendance percentage: "))
    students[name] = {"grade": grade, "attendance": attendance}
    print("Student added successfully.")
def update_student():
    name = input("Enter student name to update: ")
    if name in students:
        grade = input("Enter new grade: ")
        attendance = int(input("Enter new attendance percentage: "))
        students[name] = {"grade": grade, "attendance": attendance}
        print("Record updated successfully.")
```

```
    else:
        print("Student not found.")
def display_students():
    for name, info in students.items():
        print(f"{name}: Grade - {info['grade']}, Attendance - {info['attendance']}%")
while True:
    print("\n1. Add Student\n2. Update Student\n3. Display Students\n4. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        add_student()
    elif choice == 2:
        update_student()
    elif choice == 3:
        display_students()
    elif choice == 4:
        print("Exiting...")
        break
    else:
        print("Invalid choice. Try again.")
```

```
students = {}

def add_student():
    name = input("Enter student name: ")
    grade = input("Enter grade: ")
    attendance = int(input("Enter attendance percentage: "))
    students[name] = {"grade": grade, "attendance":
        attendance}
    print("Student added successfully.")

def update_student():
    name = input("Enter student name to update: ")
    if name in students:
        grade = input("Enter new grade: ")
        attendance = int(input("Enter new attendance
            percentage: "))
        students[name] = {"grade": grade, "attendance":
            attendance}
```

```
        print("Record updated successfully.")
    else:
        print("Student not found.")

def display_students():
    for name, info in students.items():
        print(f"{name}: Grade - {info['grade']}, Attendance -
            {info['attendance']}%")

while True:
    print("\n1. Add Student\n2. Update Student\n3. Display
        Students\n4. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        add_student()
    elif choice == 2:
        update_student()
    elif choice == 3:
        display_students()
    elif choice == 4:
        print("Exiting...")
        break
    else:
        print("Invalid choice. Try again.")
```

**Output:**