

# student\_information.py

```
def main(): # create an empty dictionary
    students = {}

    # add students
    students['Jim'] = {'id': 1, 'gpa': 3.1, 'credits-completed': 97.0,
    'grades': [80, 50, 100, 98]}
    students['Sarah'] = {'id': 2, 'gpa': 3.6, 'credits-completed': 40.0,
    'grades': [80, 98]}

    # print full dictionary
    print(students)
    print()

    # list of students
    print("List of Students")
    for name in students:
        print(name)
    print()

    # student information
    print("Student Information")
    print("Student ID\tGPA\tCredits Completed\tGrades")
    for name, info in students.items():
        print(f"{name}\t{info['id']}\t{info['gpa']}\t{info['credits-'
            completed']} \t\t{info['grades']}")
    print()

    # accessing student info using the key in a loop
    print("Accessing Student Information Using the Key in a Loop")
    for name in students:
        print(name, students[name])
    print()

    # remove a student
    print("Sarah has dropped out, removing from student info registry")
    students.pop('Sarah', None)
```

```
print(students)
print()

# getting GPA info
print("Getting Jim's GPA")
print(students.get('Jim').get('gpa'))
print()

# clear registry
print("Students have graduated, clearing the student registry")
students.clear()
print(students)
print()

print("Completed by, Matthew Valadez")

if name == "main": main()
```

IDLE Shell 3.13.7

File Edit Shell Debug Options Window Help

```
Python 3.13.7 (tags/v3.13.7:bceelc3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>>
= RESTART: C:/Users/matth/AppData/Local/Programs/Python/Python313/student_information.py
({'Jim': {'id': 1, 'gpa': 3.1, 'credits-completed': 97.0, 'grades': [80, 50, 100, 98]}, 'Sarah': {'id': 2, 'gpa': 3.6, 'credits-completed': 40.0, 'grades': [80, 98]})

List of Students
Jim
Sarah

Student Information
Student ID      GPA      Credits Completed      Grades
Jim      1      3.1      97.0          [80, 50, 100, 98]
Sarah    2      3.6      40.0          [80, 98]

Accessing Student Information Using the Key in a Loop
Jim {'id': 1, 'gpa': 3.1, 'credits-completed': 97.0, 'grades': [80, 50, 100, 98]}
}
Sarah {'id': 2, 'gpa': 3.6, 'credits-completed': 40.0, 'grades': [80, 98]}

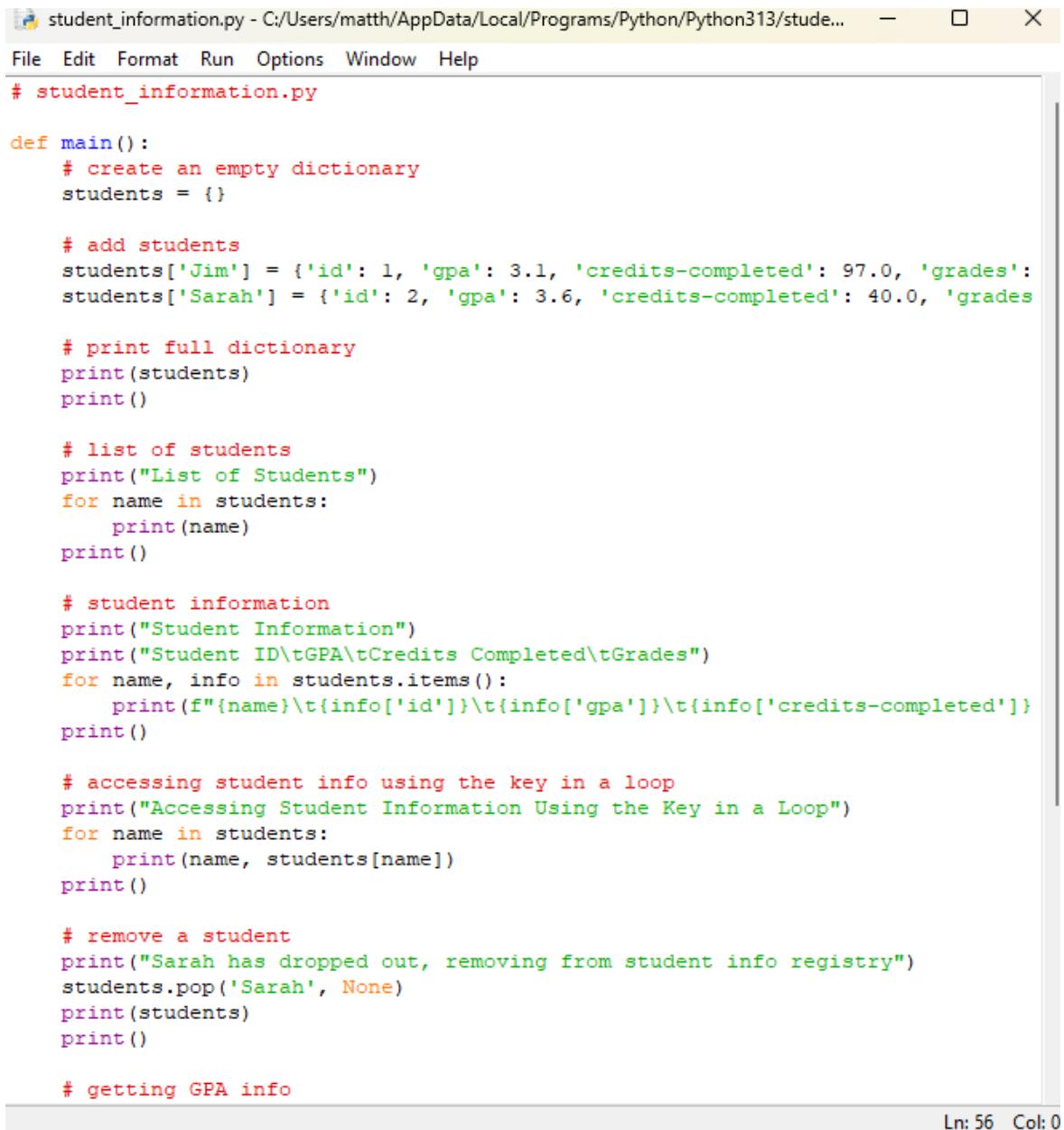
Sarah has dropped out, removing from student info registry
{'Jim': {'id': 1, 'gpa': 3.1, 'credits-completed': 97.0, 'grades': [80, 50, 100, 98]})

Getting Jim's GPA
3.1

Students have graduated, clearing the student registry
{}

Completed by, Matthew Valadez
>>> |
```

Ln: 30 Col: 0



A screenshot of a Python code editor window titled "student\_information.py - C:/Users/matt/AppData/Local/Programs/Python/Python313/stude...". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code itself is a script named "student\_information.py" which defines a function main(). This function creates an empty dictionary students, adds two students (Jim and Sarah) with their id, gpa, credits completed, and grades, prints the full dictionary, lists the students, prints student information in a tabular format, accesses student info using a loop, removes a student (Sarah), and gets GPA info.

```
# student_information.py

def main():
    # create an empty dictionary
    students = {}

    # add students
    students['Jim'] = {'id': 1, 'gpa': 3.1, 'credits-completed': 97.0, 'grades': 'A'}
    students['Sarah'] = {'id': 2, 'gpa': 3.6, 'credits-completed': 40.0, 'grades': 'B-'}

    # print full dictionary
    print(students)
    print()

    # list of students
    print("List of Students")
    for name in students:
        print(name)
    print()

    # student information
    print("Student Information")
    print("Student ID\tGPA\tCredits Completed\tGrades")
    for name, info in students.items():
        print(f"{name}\t{info['id']}\t{info['gpa']}\t{info['credits-completed']} {info['grades']}")
    print()

    # accessing student info using the key in a loop
    print("Accessing Student Information Using the Key in a Loop")
    for name in students:
        print(name, students[name])
    print()

    # remove a student
    print("Sarah has dropped out, removing from student info registry")
    students.pop('Sarah', None)
    print(students)
    print()

    # getting GPA info
```

Ln: 56 Col: 0

A screenshot of a Python code editor window titled "student\_information.py - C:/Users/mattt/AppData/Local/Programs/Python/Python313/stude...". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code itself is color-coded and defines a dictionary of student information and performs various operations on it.

```
for name in students:
    print(name)
print()

# student information
print("Student Information")
print("Student ID\tGPA\tCredits Completed\tGrades")
for name, info in students.items():
    print(f"{name}\t{info['id']}\t{info['gpa']}\t{info['credits-completed']}")
print()

# accessing student info using the key in a loop
print("Accessing Student Information Using the Key in a Loop")
for name in students:
    print(name, students[name])
print()

# remove a student
print("Sarah has dropped out, removing from student info registry")
students.pop('Sarah', None)
print(students)
print()

# getting GPA info
print("Getting Jim's GPA")
print(students.get('Jim').get('gpa'))
print()

# clear registry
print("Students have graduated, clearing the student registry")
students.clear()
print(students)
print()

print("Completed by, Matthew Valadez")

if __name__ == "__main__":
    main()
```

Ln: 56 Col: 0