Code:

```
import json

# ===== Class Definitions =====
class Person:
    def __init__(self, name, age, address):
        self.name = name
        self.age = age
        self.address = address

    def display_person_info(self):
        print(f"Name: {self.name}")
        print(f"Age: {self.age}")
        print(f"Address: {self.address}")

class Student(Person):
    def __init__(self, name, age, address, student_id):
        super().__init__(name, age, address)
        self.student_id = student_id
        self.grades = {}
        self.courses = []

    def add_grade(self, subject, grade):
        self.grades[subject] = grade

    def enroll_course(self, course):
        if course.course_name not in self.courses:
            self.courses.append(course.course_name)

    def display_student_info(self):
        print("Student Information:")
        print(f"Name: {self.name}")
        print(f"ID: {self.student_id}")
        print(f"Age: {self.age}")
        print(f"Address: {self.address}")
        print(f"Enrolled Courses: {', '.join(self.courses) if self.courses else
'None'}")
        print(f"Grades: {self.grades if self.grades else 'No grades'}")

class Course:
    def __init__(self, course_name, course_code, instructor):
        self.course_name = course_name
        self.course_code = course_code
        self.instructor = instructor
```

```
        self.students = []

    def add_student(self, student):
        if student.student_id not in [s.student_id for s in self.students]:
            self.students.append(student)

    def display_course_info(self):
        print("Course Information:")
        print(f"Course Name: {self.course_name}")
        print(f"Code: {self.course_code}")
        print(f"Instructor: {self.instructor}")
        print(f"Enrolled Students: {', '.join([s.name for s in self.students]) if
self.students else 'None'}")

# ===== Data Storage =====
students = {}
courses = {}

# ===== File Operations =====
def save_data():
    data = {
        "students": {
            sid: {
                "name": s.name,
                "age": s.age,
                "address": s.address,
                "student_id": s.student_id,
                "grades": s.grades,
                "courses": s.courses
            }
            for sid, s in students.items()
        },
        "courses": {
            cid: {
                "course_name": c.course_name,
                "course_code": c.course_code,
                "instructor": c.instructor,
                "students": [s.student_id for s in c.students]
            }
            for cid, c in courses.items()
        }
    }
    with open("data.json", "w") as f:
        json.dump(data, f, indent=4)
    print("All student and course data saved successfully.")
```

```
def load_data():
    global students, courses
    try:
        with open("data.json", "r") as f:
            data = json.load(f)

        students = {
            sid: Student(info["name"], info["age"], info["address"],
info["student_id"])
            for sid, info in data["students"].items()
        }
        for sid, info in data["students"].items():
            students[sid].grades = info["grades"]
            students[sid].courses = info["courses"]

        courses = {
            cid: Course(info["course_name"], info["course_code"],
info["instructor"])
            for cid, info in data["courses"].items()
        }
        for cid, info in data["courses"].items():
            for sid in info["students"]:
                if sid in students:
                    courses[cid].add_student(students[sid])
        print("Data loaded successfully.")
    except FileNotFoundError:
        print("No saved data found.")

# ===== Main Menu =====
def menu():
    while True:
        print("\n==== Student Management System ====")
        print("1. Add New Student")
        print("2. Add New Course")
        print("3. Enroll Student in Course")
        print("4. Add Grade for Student")
        print("5. Display Student Details")
        print("6. Display Course Details")
        print("7. Save Data to File")
        print("8. Load Data from File")
        print("0. Exit")

        choice = input("Select Option: ")
```

```
        if choice == "1":
            name = input("Enter Name: ")
            age = int(input("Enter Age: "))
            address = input("Enter Address: ")
            sid = input("Enter Student ID: ")
            if sid in students:
                print("Student ID already exists.")
            else:
                students[sid] = Student(name, age, address, sid)
                print(f"Student {name} (ID: {sid}) added successfully.")

        elif choice == "2":
            cname = input("Enter Course Name: ")
            ccode = input("Enter Course Code: ")
            instructor = input("Enter Instructor: ")
            if ccode in courses:
                print("Course code already exists.")
            else:
                courses[ccode] = Course(cname, ccode, instructor)
                print(f"Course {cname} (Code: {ccode}) created with instructor
{instructor}.")

        elif choice == "3":
            sid = input("Enter Student ID: ")
            ccode = input("Enter Course Code: ")
            if sid in students and ccode in courses:
                student = students[sid]
                course = courses[ccode]
                course.add_student(student)
                student.enroll_course(course)
                print(f"Student {student.name} (ID: {sid}) enrolled in
{course.course_name} (Code: {ccode}).")
            else:
                print("Invalid Student ID or Course Code.")

        elif choice == "4":
            sid = input("Enter Student ID: ")
            ccode = input("Enter Course Code: ")
            grade = input("Enter Grade: ")
            if sid in students and ccode in courses:
                student = students[sid]
                course = courses[ccode]
                if course.course_name in student.courses:
                    student.add_grade(course.course_name, grade)
```

```
                    print(f"Grade {grade} added for {student.name} in
{course.course_name}.")
                else:
                    print("Student is not enrolled in this course.")
            else:
                print("Invalid Student ID or Course Code.")

        elif choice == "5":
            sid = input("Enter Student ID: ")
            if sid in students:
                students[sid].display_student_info()
            else:
                print("Student not found.")

        elif choice == "6":
            ccode = input("Enter Course Code: ")
            if ccode in courses:
                courses[ccode].display_course_info()
            else:
                print("Course not found.")

        elif choice == "7":
            save_data()

        elif choice == "8":
            load_data()

        elif choice == "0":
            print("Exiting Student Management System. Goodbye!")
            break

        else:
            print("Invalid option. Please try again.")

# Run the program
if __name__ == "__main__":
    menu()
```

Output:

1.

```
==== Student Management System ====
1. Add New Student
2. Add New Course
3. Enroll Student in Course
4. Add Grade for Student
5. Display Student Details
6. Display Course Details
7. Save Data to File
8. Load Data from File
0. Exit
```

2.

```
Select Option: 1
Enter Name: Arifuzzaman
Enter Age: 25
Enter Address: Kanchan, Rupganj
Enter Student ID: 2201
Student Arifuzzaman (ID: 2201) added successfully.
```

3.

```
Select Option: 2
Enter Course Name: Data Mining
Enter Course Code: CSE 435
Enter Instructor: Atik Sir
Course Data Mining  (Code: CSE 435) created with instructor Atik Sir.
```

4.

```
Select Option: 3
Enter Student ID: 2201
Enter Course Code: CSE 435
Student Arifuzzaman (ID: 2201) enrolled in Data Mining  (Code: CSE 435).
```

5.

```
Select Option: 4
Enter Student ID: 2201
Enter Course Code: CSE 435
Enter Grade: A+
Grade A+ added for Arifuzzaman in Data Mining .
```

6.

```
Select Option: 5
Enter Student ID: 2201
Student Information:
Name: Arifuzzaman
ID: 2201
Age: 25
Address: Kanchan, Rupganj
Enrolled Courses: Data Mining
Grades: {'Data Mining ': 'A+'}
```

```
Select Option: 5
Enter Student ID: 2202
Student Information:
Name: Fahmida Akhter Orpi
ID: 2202
Age: 24
Address: Netrokona, Mymensingh
Enrolled Courses: Bangladesh Studies
Grades: {'Bangladesh Studies ': 'A+'}
```

7.

```
Select Option: 6
Enter Course Code: CSE 435
Course Information:
Course Name: Data Mining
Code: CSE 435
Instructor: Atik Sir
Enrolled Students: Arifuzzaman
```

8.

```
Select Option: 7
All student and course data saved successfully.
```