

```

import numpy as np
import matplotlib.pyplot as plt

class Student:
    def __init__(self, name, roll_number, grades):
        self.name = name
        self.roll_number = roll_number
        self.grades = grades

    def average_grade(self):
        return np.mean(self.grades)

    def grade_summary(self):
        return f"Name: {self.name}, Roll No: {self.roll_number}, Grades: {self.grades}, Average: {self.average_grade():.2f}"

class GradeSystem:
    def __init__(self):
        self.students = []

    def add_student(self, name, roll_number, grades):
        student = Student(name, roll_number, np.array(grades))
        self.students.append(student)
        print(f"Student {name} added successfully.")

    def view_students(self):
        if not self.students:
            print("No student data available.")
            return
        print("\nStudent Data:")
        for student in self.students:
            print(student.grade_summary())

    def highest_scorer(self):
        if not self.students:
            print("No student data available.")
            return
        top_student = max(self.students, key=lambda s: s.average_grade())
        print(f"\nTop Scorer: {top_student.name} with Average Grade: {top_student.average_grade():.2f}")

    def visualize_grades(self):
        if not self.students:
            print("No student data available.")
            return

        names = [student.name for student in self.students]
        averages = [student.average_grade() for student in self.students]

        plt.figure(figsize=(10, 6))
        plt.bar(names, averages, color='skyblue')
        plt.title("Student Average Grades", fontsize=16)
        plt.xlabel("Student Name", fontsize=12)
        plt.ylabel("Average Grade", fontsize=12)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()

    def save_data(self, filename):
        data = []
        for student in self.students:
            data.append([student.name, student.roll_number] + student.grades.tolist())
        np.savetxt(filename, data, fmt='%s', delimiter=',')
        print(f"Data saved to {filename}")

    def load_data(self, filename):
        try:
            data = np.loadtxt(filename, dtype=str, delimiter=',')
            self.students = []
            for row in data:
                name = row[0]
                roll_number = row[1]
                grades = list(map(float, row[2:]))
                self.add_student(name, roll_number, grades)
            print(f"Data loaded from {filename}")
        except OSError:
            print("No saved data found.")

def main():
    system = GradeSystem()

```

```
while True:
    print("\nMenu:")
    print("1. Add Student")
    print("2. View Students")
    print("3. Display Highest Scorer")
    print("4. Visualize Grades")
    print("5. Save Data")
    print("6. Load Data")
    print("7. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        name = input("Enter student name: ")
        roll_number = input("Enter roll number: ")
        grades = list(map(float, input("Enter grades separated by space: ").split()))
        system.add_student(name, roll_number, grades)
    elif choice == "2":
        system.view_students()
    elif choice == "3":
        system.highest_scorer()
    elif choice == "4":
        system.visualize_grades()
    elif choice == "5":
        filename = input("Enter filename to save data (e.g., grades.csv): ")
        system.save_data(filename)
    elif choice == "6":
        filename = input("Enter filename to load data (e.g., grades.csv): ")
        system.load_data(filename)
    elif choice == "7":
        print("Exiting the program. Goodbye!")
        break
    else:
        print("Invalid choice! Please try again.")

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