

Assignment 6 Solution

Submitted by : Yameen Ali

[Go to Github repository](#)

Student Management System Using OOP Concepts

```
In [65]: # Define a Student Class
class Student:
    total_students = 0 #class variable total_students to keep track of the total number of students.
    # Implement a constructor (__init__) to initialize the student attributes.
    def __init__(self,name,roll_number,age):
        self.name = name
        self.roll_number = roll_number
        self.age = age
        # Ensure that the marks attribute is initialized as an empty dictionary.
        self.marks = {}
        Student.total_students += 1 #Update the class variable whenever a new student object is created.

    # implement instance methods
    def add_marks(self,subject,marks):
        self.marks[subject] = marks

    def calculate_total_marks(self):
        return sum(self.marks.values())

    def calculate_percentage(self):
        if not self.marks: # Check if marks dictionary is empty
            return 0.0 # Return 0 percentage if no marks are added
        total_marks = self.calculate_total_marks()
        return (total_marks / (len(self.marks) * 100)) * 100

    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Age: {self.age}")
        print(f"Marks:")
        for subject,marks in self.marks.items():
            print(f"{subject}: {marks}")

    @classmethod
    def display_total_students(cls):
        print(f"Total Studetns: {cls.total_students}")

    def get_age(self):
        return self.age

    def set_age(self,age):
        self.age = age

    @staticmethod
    def is_valid_roll_number(roll_number):
        return isinstance(roll_number,(int,str))

    @classmethod
    def find_topper(cls, students):
        topper = None
        max_percentage = 0
        for student in students:
            percentage = student.calculate_percentage()
            if percentage > max_percentage:
                max_percentage = percentage
                topper = student
        return topper

class TopperStudent(Student):
    bonus_marks = 10

    def __init__(self,name,roll_number,age):
        super().__init__(name,roll_number,age)

    def calculate_percentage(self):
        total_marks = self.calculate_total_marks() + TopperStudent.bonus_marks
        return (total_marks / (len(self.marks) * 100)) * 100

def main():
```

```

students = [] # empty list to store student record
while True:
    print("\nStudent Management System")
    print("1. Add new Student")
    print("2. Add marks")
    print("3. Display Stduent Details")
    print("4. Display Total Students")
    print("5. Display Topper")
    print("6. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        name = input("Enter a name: ")
        roll_number = input("Enter Roll number: ")
        age = input("Enter age: ")
        student = Student(name, roll_number, age)
        students.append(student)
        print("Student added scuccesfully")

    elif choice == "2":
        roll_number = input("Enter roll number of the student")
        for student in students:
            if student.roll_number == roll_number:
                subject = input("Enter Subject name: ")
                marks = int(input("Enter marks: "))
                student.add_marks(subject, marks)
                print("Marks added successfully")
                break

    elif choice == "3":
        roll_number = input("Enter roll number of the student: ")
        for student in students:
            if student.roll_number == roll_number:
                student.display_details()
                print(f"Total Marks: {student.calculate_total_marks()}") # print total marks
                percentage = int(student.calculate_percentage())
                print(f"Percentage: {percentage}%") # Print percentage
                break
            else:
                print("Student not found.")

    elif choice == "4":
        Student.display_total_students()

    elif choice == "5":
        # Find the topper among all students
        topper = Student.find_topper(students)
        if topper:
            print("\nTopper Details:")
            topper.display_details()
            print(f"Total Marks: {topper.calculate_total_marks()}")
            print(f"Percentage: {topper.calculate_percentage()}%")
        else:
            print("No topper found.")

    elif choice == "6":
        print("Exiting...")
        break

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

if __name__ == "__main__":
    main()

```

```

Student Management System
1. Add new Student
2. Add marks
3. Display Stduent Details
4. Display Total Students
5. Display Topper
6. Exit
Student added scuccesfully

```

```

Student Management System
1. Add new Student
2. Add marks
3. Display Stduent Details
4. Display Total Students
5. Display Topper
6. Exit

```

Student added successfully

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Marks added successfully

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Marks added successfully

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Name: yameen
Roll Number: 50
Age: 22
Marks:
DSP: 85
Total Marks: 85
Percentage: 85%

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Name: Rafay
Roll Number: 18
Age: 22
Marks:
DSP: 84
Total Marks: 84
Percentage: 84%

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Total Students: 2

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Topper Details:
Name: yameen
Roll Number: 50
Age: 22
Marks:
DSP: 85
Total Marks: 85
Percentage: 85.0%

Student Management System
1. Add new Student
2. Add marks
3. Display Student Details
4. Display Total Students
5. Display Topper
6. Exit
Exiting

