

12214994_Saurabh Rana

Question No : 1 / 1

Problem Statement: Student Information Management

Objective:

Create a program to manage student information, allowing the creation of a default student and a custom student with user-provided details.

Description:

You are required to implement a program that can create and display information for students. The program should be able to create a default student with predefined attributes and also accept user input to create a new student with custom attributes. Write the solution within the `Program.cs` file.

Requirements:

1. Class Definition:

- Create a public class named **Student**.

2. Properties:

- Define three public properties:
 - **Name** (of type string): to store the student's name.
 - **Age** (of type int): to store the student's age.
 - **Grade** (of type string): to store the student's grade.

3. Constructors:

Default Constructor:

- Create a default constructor that initializes the Name, Age, and Grade properties with the following default values:
 - **Name = "Hellen Doe"**
 - **Age = 21**

- Grade = "A"

Parameterized Constructor:

- Create a parameterized constructor that allows initialization of the Name, Age, and Grade properties using values provided as arguments.
-

4. Main Program:

- Create an instance of the Student class using the default constructor and display the default student's details (name, age, and grade).
 - Prompt the user to input a student's name, age, and grade.
 - Create another instance of the Student class using the parameterized constructor with the input values and display the new student's details (name, age, and grade).
-

Input Format:

The program should prompt the user to enter the following information for the new student:

- **Name:** A string representing the student's name.
 - **Age:** An integer representing the student's age.
 - **Grade:** A string representing the student's grade.
-

Output Format:

- The details of the default student.
 - The details of the new student created with user input.
-

Sample Input 1:

John Doe
18
A

Sample Output 1:

Default Student:
Name: Hellen Doe
Age: 21

Grade: A
New Student:
Name: John Doe
Age: 18
Grade: A

Sample Input 2:

James
45
C

Sample Output 2:

Default Student:
Name: Hellen Doe
Age: 21
Grade: A
New Student:
Name: James
Age: 45
Grade: C

Commands to Run the Project:

- cd dotnetapp
(Select the dotnet project folder)
 - dotnet run
(To run the application)
 - dotnet build
(To build and check for errors)
 - dotnet clean
(If the same error persists, clean the project and build again)
-

Note:

The project will not be submitted if "**Submit Project**" is not done at least once.

Answer:

```
class Student  
{
```

```
public string Name { get; set; }

public int Age { get; set; }

public string Grade { get; set; }

// Create a default constructor that initializes the Name, Age, and Grade properties

public Student()

{

    Name = "Hellen Doe";

    Age = 21;

    Grade = "A";

}

// Create a parameterized constructor that allows initialization of the Name, Age, and Grade properties using values provided as arguments.

public Student(string name, int age, string grade)

{

    Name = name;

    Age = age;

    Grade = grade;

}

class Program

{

    public static void Main()

    {

        Student student1 = new Student();
```

```
Console.WriteLine("Student 1:");  
Console.WriteLine("Name: " + student1.Name);  
Console.WriteLine("Age: " + student1.Age);  
Console.WriteLine("Grade: " + student1.Grade);  
  
Console.WriteLine("\nEnter details for Student 2:");  
Console.Write("Name: ");  
string name = Console.ReadLine() ?? "";  
Console.Write("Age: ");  
int age = Convert.ToInt32(Console.ReadLine());  
Console.Write("Grade: ");  
string grade = Console.ReadLine() ?? "";  
  
Student student2 = new Student(name, age, grade);  
Console.WriteLine("\nStudent 2:");  
Console.WriteLine("Name: " + student2.Name);  
Console.WriteLine("Age: " + student2.Age);  
Console.WriteLine("Grade: " + student2.Grade);  
  
}
```