Inheritance

by AllisonP

Objective

Today, we're delving into Inheritance. Check out the Tutorial tab for learning materials and an instructional video!

Task

You are given two classes, *Person* and *Student*, where *Person* is the base class and *Student* is the derived class. Completed code for *Person* and a declaration for *Student* are provided for you in the editor. Observe that *Student* inherits all the properties of *Person*.

Complete the *Student* class by writing the following:

- A *Student* class constructor, which has 4 parameters:
  1. A string, $firstName$.
  2. A string, $lastName$.
  3. An integer, id.
  4. An integer array (or vector) of test scores, $scores$.
- A *char calculate()* method that calculates a Student object's average and returns the grade character representative of their calculated average:

Grading Scale

Letter	Average (a)
O	$90 \le a \le 100$
E	$80 \le a < 90$
A	$70 \le a < 80$
P	$55 \le a < 70$
D	$40 \le a < 55$
T	$a < 40$

Input Format

The locked stub code in your editor calls your *Student* class constructor and passes it the necessary arguments. It also calls the *calculate* method (which takes no arguments).

You are not responsible for reading the following input from stdin:
The first line contains $firstName$, $lastName$, and id, respectively. The second line contains the number of test scores. The third line of space-separated integers describes $scores$.

Constraints

- $4 \le |firstName|, |lastName| \le 10$
- $|id| \equiv 7$
- $0 \le score, average \le 100$

Output Format

This is handled by the locked stub code in your editor. Your output will be correct if your *Student* class constructor and *calculate()* method are properly implemented.

Sample Input

```
Heraldo Memelli 8135627
2
100 80
```

Sample Output

```
Name: Memelli, Heraldo
ID: 8135627
Grade: O
```

Explanation

This student had 2 scores to average: 100 and 80. The student's average grade is $\frac{(100+80)}{2} = 90$. An average grade of 90 corresponds to the letter grade O, so our *calculate()* method should return the character `'O'` .

Submissions: 186
Max Score: 30
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Linq;

class Person{
    protected string firstName;
    protected string lastName;
    protected int id;

    public Person(){}
    public Person(string firstName, string lastName, int identification){
            this.firstName = firstName;
            this.lastName = lastName;
            this.id = identification;
    }
    public void printPerson(){
        Console.WriteLine("Name: " + lastName + ", " + firstName + "\nID: " + id);
    }
}

class Student : Person{
    private int[] testScores;

            /*
            *   Class Constructor
            *
            *   Parameters:
            *   firstName - A string denoting the Person's first name.
            *   lastName - A string denoting the Person's last name.
            *   id - An integer denoting the Person's ID number.
            *   scores - An array of integers denoting the Person's test scores.
            */
            // Write your constructor here

            public Student(string firstName, string lastName, int identification, int[] scores)
                : base(firstName, lastName, identification)
            {
                this.testScores = scores;
            }
            /*
            *   Method Name: Calculate
            *   Return: A character denoting the grade.
            */
            // Write your method here
```

```
            public char Calculate()
            {
                int a = this.testScores.Sum() / this.testScores.Length;
                if (90 <= a && a <= 100)
                {
                    return 'O';
                }
                else if (80 <= a && a < 90)
                {
                    return 'E';
                }
                else if (70 <= a && a < 80)
                {
                    return 'A';
                }
                else if (55 <= a && a < 70)
                {
                    return 'P';
                }
                else if (40 <= a && a < 55)
                {
                    return 'D';
                }
                return 'T';
            }
}
class Solution {
    static void Main() {
        string[] inputs = Console.ReadLine().Split();
        string firstName = inputs[0];
        string lastName = inputs[1];
        int id = Convert.ToInt32(inputs[2]);
        int numScores = Convert.ToInt32(Console.ReadLine());
        inputs = Console.ReadLine().Split();
        int[] scores = new int[numScores];
        for(int i = 0; i < numScores; i++){
            scores[i]= Convert.ToInt32(inputs[i]);
        }

        Student s = new Student(firstName, lastName, id, scores);
        s.printPerson();
        Console.WriteLine("Grade: " + s.Calculate() + "\n");
    }
}
```

Line: 68 Col: 14

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7

Next Challenge