









Rank

Leaderboard







All Domains > Tutorials > 30 Days of Code > Day 4: Class vs. Instance

# Day 4: Class vs. Instance

Submissions



Problem

Leaderboard

Discussions

Editorial

Tutorial

#### **Objective**

In this challenge, we're going to learn about the difference between a class and an instance; because this is an Object Oriented concept, it's only enabled in certain languages. Check out the Tutorial tab for learning materials and an instructional video!

#### Task

Write a Person class with an instance variable, age, and a constructor that takes an integer, initial Age, as a parameter. The constructor must assign initial Age to age after confirming the argument passed as initial Age is not negative; if a negative argument is passed as initial Age, the constructor should set age to 0 and print Age is not valid, setting age to 0. In addition, you must write the following instance methods:

1. yearPasses() should increase the age instance variable by 1.

2. amlOld() should perform the following conditional actions:

- If age < 13, print You are young...
- If  $age \geq 13$  and age < 18, print You are a teenager...
- Otherwise, print You are old..

To help you learn by example and complete this challenge, much of the code is provided for you, but you'll be writing everything in the future. The code that creates each instance of your Person class is in the main method. Don't worry if you don't understand it all quite yet!

Note: Do not remove or alter the stub code in the editor.

#### **Input Format**

Input is handled for you by the stub code in the editor.

The first line contains an integer, T (the number of test cases), and the T subsequent lines each contain an integer denoting the age of a Person instance.

#### Constraints

- $1 \le T \le 4$
- $-5 \le age \le 30$

#### **Output Format**

Complete the method definitions provided in the editor so they meet the specifications outlined above; the code to test your work is already in the editor. If your methods are implemented correctly, each test case will print 2 or 3 lines (depending on whether or not a valid initial Age was passed to the constructor).

# Sample Input

- -1 10
- 16

#### Sample Output

```
Age is not valid, setting age to 0.
You are young.
You are young.
You are young.
You are a teenager.
You are old.
You are old.
You are old.
```

## **Explanation**

### Test Case 0: initialAge = -1

Because initialAge < 0, our code must set age to 0 and print the "Age is not valid..." message followed by the young message. Three years pass and age = 3, so we print the young message again.

# Test Case 1: initialAge = 10

Because initialAge < 13, our code should print that the person is young. Three years pass and age = 13, so we print that the person is now a teenager.

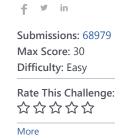
# Test Case 2: initialAge = 16

Because  $13 \le initialAge < 18$ , our code should print that the person is a teenager. Three years pass and age = 19, so we print that the person is old.

#### Test Case 3: initialAge = 18

Because  $initialAge \ge 18$ , our code should print that the person is old. Three years pass and the person is still old at age = 21, so we print the old message again.

The extra line at the end of the output is supposed to be there and is trimmed before being compared against the test case's expected output. If you're failing this challenge, check your logic and review your print statements for spelling errors.



```
Current Buffer (saved locally, editable) & 🗘
                                                                                      C#
    using System;
 1
    using System.Collections.Generic;
 2
 3
    using System.IO;
 4
 5
 6
     class Person {
 7
         public int age;
 8
             public Person(int initialAge)
 9
10
                  // Add some more code to run some checks on initialAge
11
                  if (initialAge < 0)</pre>
12
13
                      age = 0:
14
                      Console.WriteLine("Age is not valid, setting age to 0.");
15
16
                  else
17
                  {
                      age = initialAge:
18
19
20
```

```
public void amIOld()
21
22 •
23
                  // Do some computations in here and print out the correct statement to the console
24
                 if (age < 13)
25
                      Console.WriteLine("You are young.");
26
27
28
                 else if (age >= 13 && age < 18)
29
30
                      Console.WriteLine("You are a teenager.");
31
32
                 else
33
                  {
34
                      Console.WriteLine("You are old.");
35
36
37
38
             public void yearPasses()
39
40
                  // Increment the age of the person in here
41
                 this age++;
42
43 ▼ static void Main(String[] args) {
44
            int T=int.Parse(Console.In.ReadLine());
45 ▼
            for (int i = 0; i < T; i++) {
46
                int age=int.Parse(Console.In.ReadLine());
47
               Person p=new Person(age);
48
                p.amIOld();
49 ▼
                   for (int j = 0; j < 3; j++) {
50
                     p.yearPasses();
51
                   p.amIOld();
52
53
                   Console.WriteLine();
54
           }
55
        }
56
   }
57
                                                                                                           Line: 53 Col: 37
```

Run Code

Submit Code

# Congrats, you solved this challenge! ✓ Test Case #0 ✓ Test Case #1 ✓ Next Challenge

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature