

# CS 112, Foundations of CS

## lab 7: Classes – Inheritance

### Submit to Canvas

Computer  
Science

This lab is worth 100 points. The goal for this lab is:

#### I. Writing multiple classes and using inheritance

If you do not complete the lab in the time allotted, then please return to the lab in your spare time, and complete it by the **due date, which is specified on Canvas**.

#### Preliminaries

- For this lab, create another folder, called **lab7**.
- Navigate to the *winpython* folder on your computer. The path to the folder is the following: **C:\bin\winpython**. If you need help getting to the folder please refer back to lab 1. Once inside the *winpython* folder double click on “Spyder” icon and it will open up the Spyder IDE.

#### I. Writing multiple classes and using inheritance

For this programming task you will write a base class “Human” and a few subclasses such as “Swimmer” and “Hiker”. Classes “Swimmer” and “Hiker” will inherit from the class “Human”.

1. To get started create a new file and save it as **People.py**.
2. In the comment section on top, include your name, today's date and program name (lab 7, People.py).
3. Create a class and call it **Human**, it should take in two attributes, name and age. Write the “**\_\_init\_\_()**” method and two additional methods. A method named “**hobby()**”, which prints out a generic message such as “Likes watching Netflix”. Write another method named “**info()**”, this method should print out the “**name**” and “**age**”. In total you should have three methods in the “**Human**” class.
4. Write another class and name it “**Hiker**”. This class should inherit from the “**Human**” class. This class should only have one method, which overrides the “**hobby()**” method to print “Likes going on hikes”.
5. Write another class and name it “**Scientist**”, it should also inherit from the “**Human**” class. The “**Scientist**” class should take in an additional parameter named “**lab**”. Inside this class write an “**\_\_init\_\_()**” method which calls the “**\_\_init\_\_()**” method of the “**Human**” class to handle the “**name**” and “**age**”. The attribute “**lab**” should be assigned to the class “**Scientist**”.

Override the “**hobby()**” method in a similar fashion as you did for the “**Hiker**” class.

Write another method named **“labName()”**, which prints out the message “Works at the **X** laboratory”. **X** refers to the variable name **“lab”**.

6. Write another class which inherits from **“Human”**, and name it **“Swimmer”**. This class should have an additional attribute named **“hours”**. Similar to the **“Hiker”** class, define the **“\_\_init\_\_()”** method.

Override the **“hobby()”** method to print “Likes swimming in the lake”.

Write another method named **“hoursSwimming()”**. This method should print “Swims **X** hours per week”. **X** is a placeholder for the attribute **“hours”**.

7. Lastly, write another class and name it **“ScientificSwimmer”**, which inherits from the class **“Scientist”** and the class **“Swimmer”**. This class only has the **“\_\_init\_\_()”** method, which calls its parent **“\_\_init\_\_()”** methods.
8. Create an instance of each class (Human, Hiker, Scientist, Swimmer and ScientificSwimmer). On each instance call the **“info()”** and **“hobby()”** methods. Additionally call the **“labName()”** method on an instance of Scientist, the method **“hoursSwimming()”** on an instance of Swimmer. Lastly call all of the methods mentioned above on an instance of ScientificSwimmer.

**Sample output can be seen below:**

```
Megan is 20 years old.  
Likes watching Netflix
```

```
Jack is 43 years old.  
Likes going on hikes.
```

```
Mike Flex is 27 years old.  
Likes doing scientific experiments.  
Works at the research labratory.
```

```
Tom Accer is 23 years old.  
Likes swimming in the lake.  
Swims 15 hours per week.
```

```
John Smith is 30 years old.  
Likes doing scientific experiments.  
Works at the nuclear labratory.  
Swims 100 hours per week.
```

## Rubric

Upload your source code to Canvas. Here's what we are looking for, when grading your submission

The .py file must be thoroughly commented. If your code breaks (crashes) because you've been unable to fix a syntax error, then the comments will allow you to receive partial credit.

For this lab, make sure that the following file is uploaded to Canvas:

*People.py*

File / task	Points
I. <b>People.py</b>	100
Total	100