Assignment 11 – Inheritance (Humans vs Vampires)

Goals

- Practice writing classes with attributes and methods
- Creating and using instance variables
- Creating classes by using inheritance

Overview

In this assignment you will be modeling both humans and vampires by using inheritance. In the main program you will simulate the interactions between a vampire and humans. You will need to separate all 4 portions (3 classes, 1 main program) into their own separate files.

Requirements

- Create a new Python project named **`ITP115_a#_lastname_firstname`** (replace ***`lastname`*** with your last/family name and ***`firstname`*** with your first name).
- This project will be a folder containing the files you create for this assignment.

Part 1: Being Class

Start by defining a **`Being`** class in a file called **`Being.py`**. This will be the parent class that the **`Human`** and **`Vampire`** classes will inherit from. Model your class using the following requirements:

- Instance Attributes
  - Define the following instance attributes. Be sure to make all attributes private.
    - **`__name`**: a string representing the being's name
    - **`__quarts`**: an integer representing the number of quarts of blood the being has
- Instance Methods
  - Define the following methods for the **`Being`** class:
    - **`__init__`** Constructor
      - Inputs: (2) the name of the Being (string), the quarts of blood the being has (integer)
      - Return value: none
      - Set the instance attributes to the values of the corresponding input parameters.
    - Getters/Setters
      - Define a get and set method for each instance attribute. You should have in total 2 get methods and 2 set methods.
    - **`increaseQuarts()`**
      - Inputs: none
      - Return value: none
      - Increase the being's quarts of blood by 1.
    - **`decreaseQuarts()`**
      - Inputs: none
      - Return value: none
      - Decrease the being's quarts of blood by 1.

Part 2: Human Class

After creating the **`Being`** class, define the **`Human`** class in a file called **`Human.py`**. The **`Human`** class needs to inherit from **`Being`**. Be sure to include the proper import statement. Model your class using the following requirements:

- Instance Attribute
  - Define the following instance attribute. Make the attribute private.
    - **`__bloodType`**: a string representing the Human's blood type
- Instance Methods
  - Define the following methods for the **`Human`** class:
    - **`__init__`** #Constructor method
      - Inputs: (3) name of the human (string), quarts of blood the human starts with (integer), blood type (string)
      - Return value: none
      - Call the parent constructor to set the name and quarts. Set the instance attribute of the **`Human`** class.
    - Getter/Setter
      - Define a get and set method for the blood type attribute.
    - **`isAlive()`**
      - Inputs: none
      - Return value: a **`boolean`**
      - Determine if the human is still alive by checking if they have more than 0 quarts of blood (i.e. a human is dead if they have 0 quarts of blood). Return **`True`** or **`False`** accordingly.
    - **`__str__`**
      - Inputs: none
      - Return value: a **`string`**
      - Construct a message about the human. The format should be similar to:
        `Human Buffy has 4 quarts of type A- blood.`

Part 3: Vampire Class

Define the **`Vampire`** class in a file called **`Vampire.py`**. The **`Vampire`** class needs to inherit from **`Being`**. Be sure to include the proper import statements. Model your class using the following requirements:

- Class Attributes
  - Define 2 class attributes for the **`Vampire`** class:
    - **`MAX_BLOOD`**
      - **`int`** that represents the maximum number of quarts of blood that a being can have. Set this value to 5 (integer).
    - **`HUNGER_LEVELS`**
      - Hunger levels is a **`list`** of **`strings`** containing the possible hunger levels of the vampire:
        "starving", "hangry", "hungry", "content", "full", "stuffed"
- Instance Attribute
  - Define the following instance attribute. Make the attribute private.
    - **`__animalForm`**: a string representing the Vampire's animal form (ex: "Bat", "Wolf")
- Instance Methods
  - Define the following methods for the **`Human`** class:
    - **`__init__`** #Constructor method
      - Inputs: (3) the vampire's name (string), the vampire's quarts of blood (integer), the vampire's animal form (string)
      - Call the parent constructor to set the name and quarts. Set the instance attribute of the **`Vampire`** class.
    - Getter/Setter
      - Define a get and set method for the animal form attribute.
    - **`getHunger()`**
      - Inputs: none
      - Return value: a **`string`** representing the vampire's hunger status
      - Use the vampire's current quarts of blood as an index for the **`HUNGER_LEVELS`** list to determine the hunger level.
    - **`isStuffed()`**
      - Inputs: none
      - Return value: a **`boolean`**

        - Determine if the **`Vampire`** is full or not by checking if its quarts of blood are equal to the max (use the class attribute to do this error check). Return **`True`** or **`False`** accordingly.
- **`suckBlood()`**
    - Inputs (1): a **`Human`** object
    - Return value: none
    - Simulate the vampire sucking blood from the human by calling **`decreaseQuarts`** method on the **`Human`** object and the vampire's **`increaseQuarts`** method.
- **`__str__`**
    - Inputs: none
    - Return value: a string
    - Construct a message about the vampire. Use the **`getName`** method when forming the message. The format should be similar to:
      `Count Dracula is hungry`

Part 4: Main

Define the main portion of your program in a separate file called **`Game.py`**. Be sure to include the proper import statements. Define functions using the following requirements:

- **`printHumans(humanList)`**
  - Inputs (1): a **`list`** of **`Human`** objects
  - Return value: none
  - Use a for loop using **`range`** access to access each **`Human`** object in the **`list`** and display a number. Do **not** hard code this menu.
- **`performFeeding(humanList, vamp)`**
  - Inputs (2): a **`list`** of **`Human`** objects, one **`Vampire`** object
  - Return value: none
  - Ask the user to enter an index number of a human. Be sure to include error checking.
  - Determine if the vampire is able to suck blood from the human.
  - **Important**:
    - A human must not be allowed to have less than zero quarts of blood, and a vampire cannot have a quarts of blood greater than five.
    - Your program should not allow vampire to suck blood from a human who has 0 quarts of blood. A stuffed vampire should not be able to suck blood from a human either.
    - You should be making use of the **`isAlive`** and **`isStuffed`** methods in the **`Human`** and **`Vampire`** classes.
  - Call the **`suckBlood`** method on the **`Vampire`** using the selected **`Human`** **<u>if</u>** you have determined that this is valid and print out all relevant information.
    - If you were unable to call **`suckBlood`**, print out a message that indicates to the user why (i.e. if the vampire if stuffed or if the human is dead).
- **`main()`**
  - Start by creating a list of human objects. You may hard code in the information about the humans (i.e. you do not need to ask the user for any input about this information). The list should contain at least 3 objects.
  - Create a vampire object. Ask the user for their name and for an animal. Use 0 for the vampire's quarts of blood (initially it is "starving").
  - Main should make use of the two methods, printHumans and performFeeding, listed above.

    - Print a menu with 3 options (print all humans, suck blood, and quit) to the user. Allow the user to continue selecting menu options until they decide to exit. Include error checking.
    - Upon exiting, print out a message about the vampire and which animal form it transforms into.
- Call the main() function.
- Please include comments.

Sample Output

```
Please enter your name: Count Dracula
Please enter an animal: bat

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 9
Error, please try again!

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 1

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 3 quarts of type B- blood.

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 3 quarts of type B- blood.
Please select a human from the list: 9

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 3 quarts of type B- blood.
Error, please try again.
Please select a human from the list: 3
Human Annie Sawyer has 2 quarts of type B- blood.
Count Dracula is hangry
```

```
Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 2 quarts of type B- blood.
Please select a human from the list: 3
Human Annie Sawyer has 1 quarts of type B- blood.
Count Dracula is hungry

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 1 quarts of type B- blood.
Please select a human from the list: 3
Human Annie Sawyer has 0 quarts of type B- blood.
Count Dracula is content

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 0 quarts of type B- blood.
Please select a human from the list: 3
Human Annie Sawyer has 0 quarts of type B- blood.
Annie Sawyer is dead! Cannot suck blood.
Count Dracula is content
```

```
Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 5 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 0 quarts of type B- blood.
Please select a human from the list: 0
Human Eric Brooks has 4 quarts of type A- blood.
Count Dracula is full

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 4 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 0 quarts of type B- blood.
Please select a human from the list: 0
Human Eric Brooks has 3 quarts of type A- blood.
Count Dracula is stuffed

Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: 2

0) Human Eric Brooks has 3 quarts of type A- blood.
1) Human Mina Harker has 7 quarts of type O+ blood.
2) Human Louis de Pointe du Lac has 4 quarts of type O+ blood.
3) Human Annie Sawyer has 0 quarts of type B- blood.
Please select a human from the list: 0
Human Eric Brooks has 3 quarts of type A- blood.
Count Dracula is stuffed
Count Dracula cannot suck any more blood!
```

```
Menu:
1 --> Print All Humans
2 --> Suck Blood
-1 --> Quit
Enter your choice: -1
Count Dracula transformed back into a bat
```

Deliverables and Submission Instructions

- Compress your Python project folder which contains your Python source code. It should contain the following files: Being.py, Human.py, Vampire.py, and Game.py.
- You should have a zip file entitled **`ITP115_a#_lastname_firstname.zip`** where you replace ***`lastname`*** with your last/family name and ***`firstname`*** with your first name.
- Under Assignments on Blackboard, upload the zip file.

Grading

Item	Points
Humans and Vampires	40
Part 1: Being Class (Being.py)	10
Part 2: Human Class (Human.py)	10
Part 3: Vampire Class (Vampire.py)	10
Part 4: Main Program (Game.py)	10
Total*	**40**

** Points will be deducted for poor code style, or improper submission.*