|  |  |
| --- | --- |
| **NAME** | **DUE** |
| **TOMOKI KOIKE** | **OCT 27, 2019** |

**Every Boiler Engineering Code – Intermediate Level Programming**

**Week 6 – Programming Exercises**

|  |  |
| --- | --- |
| **QUESTION** | **SCORE** |
| **#1** | **/10** |
| **#2** | **/15** |
| **#3** | **/30** |
| **TOTAL** | **/55** |

1. **(10 points, Class Basic)** Write a class named Pet, which should have the following data attributes:

* \_ \_name (for the name of a pet)
* \_ \_animal\_type (for the type of animal that a pet is. Example values are ‘Dogs’, ‘Cat’, and ‘Bird’)
* \_ \_age (for the pet’s age)

The **Pet** class should have an \_ \_init\_ \_method that creates these attributes. It should also have the following methods:

* set\_name

This method assigns a value to the \_ \_name field.

* set\_animal\_type

This method assigns a value to the \_ \_animal\_type field

* set\_age

This method assigns a value to the \_ \_age field

* get\_name

This method returns the value of the \_ \_name field

* get\_animal\_type

This method returns the value of the \_ \_animal\_type field

* get\_age

This method returns the value of the \_ \_age field

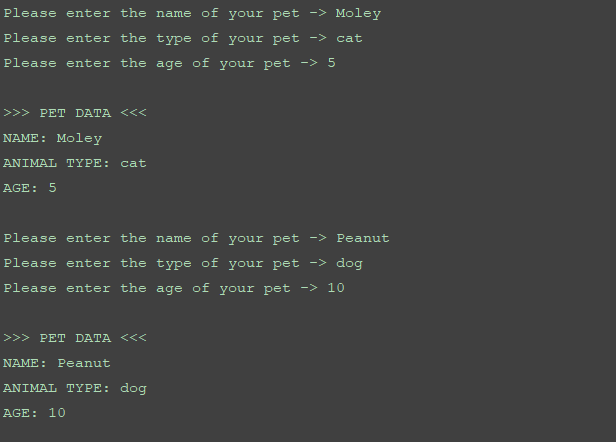
Once you have written the class, write a program that creates an object of the **Pet** class and prompts the user to enter the name, type, and age of his or her pet. This data should be stored as the object’s attributes. Use the object’s accessor methods to retrieve the pet’s name, type, and age and display this data on the screen.

**(use ‘Moley’, ‘cat’, ‘5; ‘Peanut’, ‘dog’, ‘10’ to test your code.)**

**>>CODE**

### PROBLEM 1  
# AUTHOR: Tomoki Koike  
# DATE: OCT 27 2019  
# DESCRIPTION: This program will create a class called Pet and execute the functions in  
# in the class to see that they are working fine.  
###  
  
# Main function  
**def main():** # Pet 1  
 **[**name1, animal\_type1, age1**] =** enterInputs**()** # Creating an object from the class  
 pet1 **=** Pet**(**name1, animal\_type1, age1**)** # Printing out the data of pet  
 print**()** print**('>>> PET DATA <<<')** print**('NAME: {0}'**.format**(**pet1.get\_name**()))** print**('ANIMAL TYPE: {0}'**.format**(**pet1.get\_animal\_type**()))** print**('AGE: {0}'**.format**(**pet1.get\_age**()))** print**()** # Pet 2  
 **[**name2, animal\_type2, age2**] =** enterInputs**()** # Creating an object from the class  
 pet2 **=** Pet**(**name2, animal\_type2, age2**)** # Printing out the data of pet  
 print**()** print**('>>> PET DATA <<<')** print**('NAME: {0}'**.format**(**pet2.get\_name**()))** print**('ANIMAL TYPE: {0}'**.format**(**pet2.get\_animal\_type**()))** print**('AGE: {0}'**.format**(**pet2.get\_age**()))**# Class  
**class Pet():** # Data attributes  
 **def \_\_init\_\_(**self, name, animal\_type, age**):** self.\_\_name **=** name  
 self.\_\_animal\_type **=** animal\_type  
 self.\_\_age **=** age  
  
 # Mutators  
 **def set\_name(**self, name**):** self.\_\_name **=** name  
 **def set\_animal\_type(**self, animal\_type**):** self.\_\_animal\_type **=** animal\_type  
 **def set\_age(**self, age**):** self.\_\_age **=** age  
  
 # Accessors  
 **def get\_name(**self**):  
 return** self.\_\_name  
 **def get\_animal\_type(**self**):  
 return** self.\_\_animal\_type  
 **def get\_age(**self**):  
 return** self.\_\_age  
  
# Function  
# Obtaining inputs from the user  
**def enterInputs():** name **=** input**('Please enter the name of your pet -> ')** animal\_type **=** input**('Please enter the type of your pet -> ')** age **=** input**('Please enter the age of your pet -> ')  
 return** name, animal\_type, age  
  
# Executing the main function  
**if** \_\_name\_\_ **== '\_\_main\_\_':** main**()**

**>>OUTPUT**



1. **(15 points, Class Basic)** Write a class named Car that has the following data attributes:
   1. \_ \_year\_model (for the car’s year model)
   2. \_ \_make (for the make of the car)
   3. \_ \_speed (for the car’s current speed)

The **Car** class should have an \_ \_init\_ \_ method that accepts that car’s year model and make as arguments. These values should be assigned to the object’s \_ \_year\_model and \_ \_make data attributes. It should also assign 0 to the \_ \_ speed data attribute.

The class should also have the following methods:

* accelerate

The accelerate method should **add** 5 from the speed data attribute each time it is called.

* brake

The brake method should **subtract** 5 from the speed data attribute each time it is called.

* get\_speed

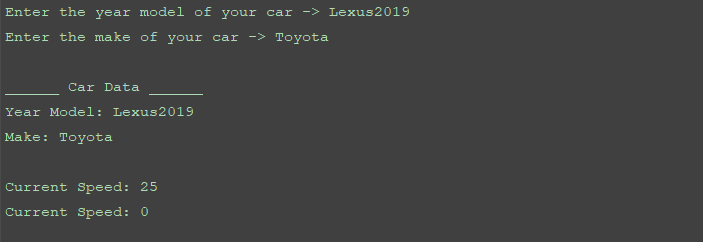
The get\_speed method should return the current speed.

Next, design a program that creates a **Car** object then calls the **accelerate** method five times. After each call to the **accelerate** method, get the current speed of the car and display it. Then call the **brake** method five times. After each call to the **brake** method, get the current speed of the car and display it.

**>>CODE**

### PROBLEM 2  
# AUTHOR: Tomoki Koike  
# DATE: OCT. 27 2019  
# DESCRIPTION: This program creates an object called car and displays the user input and  
# computes the accelerated and decelerated speeds  
###  
  
# Main function  
**def main():** # Accept user input  
 **[**year\_model, make**] =** getInputs**()** # Creating the object  
 lexus **=** car**(**year\_model, make**)** # Display the car data  
 print**()** print**('\_\_\_\_\_\_ Car Data \_\_\_\_\_\_')** print**('Year Model: {0}'**.format**(**lexus.get\_year\_model**()))** print**('Make: {0}'**.format**(**lexus.get\_make**()))** print**()** # Call the methods in the object  
 **for** x **in** range**(**5**):** lexus.accelerate**()** # Get the speed  
 print**('Current Speed: {0}'**.format**(**lexus.get\_speed**()))** # Call the brake method  
 **for** x **in** range**(**5**):** lexus.brake**()** # Get the speed  
 print**('Current Speed: {0}'**.format**(**lexus.get\_speed**()))**# Classes  
**class car():** # Data attributes  
 **def \_\_init\_\_(**self, year\_model, make**):** self.\_\_year\_model **=** year\_model  
 self.\_\_make **=** make  
 self.\_\_speed **=** 0  
  
 # Accessor  
 **def get\_year\_model(**self**):  
 return** self.\_\_year\_model  
 **def get\_make(**self**):  
 return** self.\_\_make  
 **def get\_speed(**self**):  
 return** self.\_\_speed  
  
 # Methods  
 **def accelerate(**self**):** self.\_\_speed **+=** 5  
 **return** self.\_\_speed  
 **def brake(**self**):** self.\_\_speed **-=** 5  
 **return** self.\_\_speed  
  
# Function  
# Function to get user input  
**def getInputs():** year\_model **=** input**('Enter the year model of your car -> ')** make **=** input**('Enter the make of your car -> ')  
 return** year\_model, make  
  
# Executing the main function  
**if** \_\_name\_\_ **== '\_\_main\_\_':** main**()**

**>>OUTPUT**



1. **(30 points, Class Basic) Employee**, **ProductionWorker** **and ShiftSupervisor Classes**

Write an **Employee** class that keeps data attributes for the following pieces of information:

* Employee name
* Employee number

Next, write a class named **ProductionWorker** that is a subclass of the **Employee** class. The **ProductionWorker** class should keep data attributes for the following information:

* Shift number (an integer, such as 1, 2 or 3)
* Hourly pay rate

The workday is divided into two shifts: day and night. The shift attribute will hold an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2.

Next, write a class named **ShiftSupervisor Class** that is also a subclass of the **Employee** class.In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. The **ShiftSupervisor** class should keep a data attribute for the annual salary, and a data attribute for the annual production bonus that a shift supervisor has earned.

Write the appropriate accessor and mutator method for each class.

Once you have written all the classes, write a separate program that creates objects of **ProductionWorker** class and **ShiftSupervisor** class, and prompts the user to enter data for each of the object’s data attributes (you need import the module which includes the definitions of the classes in this program). Store the data in the objects, then use the objects’ accessor methods to retrieve them and display them on the screen.

**>>CODE**

**Object file**

# This is the python file with the classes that have the  
# classes that we are instructed to create  
  
**class Employee:** # Data attributes  
 **def \_\_init\_\_(**self, EmployeeName, EmployeeNum**):** self.\_\_EmployeeName **=** EmployeeName  
 self.\_\_EmployeeNum **=** EmployeeNum  
  
 # Mutators  
 **def set\_EmployeeName(**self, EmployeeName**):** self.\_\_EmployeeName **=** EmployeeName  
 **def set\_EmployeeNum(**self, EmployeeNum**):** self.\_\_EmployeeNum **=** EmployeeNum  
  
 # Accessors  
 **def get\_EmployeeName(**self**):  
 return** self.\_\_EmployeeName  
 **def get\_EmployeeNum(**self**):  
 return** self.\_\_EmployeeNum  
  
# Subclass for production employee  
**class ProductionWorker(**Employee**):** # Data attriubtes  
 **def \_\_init\_\_(**self, shiftNum, hr\_pay\_rate, EmployeeName, EmployeeNum**):** Employee.\_\_init\_\_**(**self, EmployeeName, EmployeeNum**)** self.\_\_shiftNum **=** shiftNum  
 self.\_\_hr\_pay\_rate **=** hr\_pay\_rate  
  
 # Mutators  
 **def set\_shiftNum(**self, shiftNum**):** self.\_\_shiftNum **=** shiftNum  
 **def set\_hr\_pay\_rate(**self, hr\_pay\_rate**):** self.\_\_hr\_pay\_rate **=** hr\_pay\_rate  
  
 # Accessor  
 **def get\_shiftNum(**self**):  
 return** self.\_\_shiftNum  
 **def get\_hr\_pay\_rate(**self**):  
 return** self.\_\_hr\_pay\_rate  
  
# Subclass for shift supervisor  
**class ShiftSupervisor(**Employee**):** # Data attributes  
 **def \_\_init\_\_(**self, yearSal, yearBonus, EmployeeName, EmployeeNum**):** Employee.\_\_init\_\_**(**self, EmployeeName, EmployeeNum**)** self.\_\_yearSal **=** yearSal  
 self.\_\_yearBonus **=** yearBonus  
  
 # Mutator  
 **def set\_yearSal(**self, yearSal**):** self.\_\_yearSal **=** yearSal  
 **def set\_yearBonus(**self, yearBonus**):** self.\_\_yearBonus **=** yearBonus  
  
 # Accessor  
 **def get\_yearSal(**self**):  
 return** self.\_\_yearSal  
 **def get\_yearBonus(**self**):  
 return** self.\_\_yearBonus

**Execution file**

### PROBLEM 3  
# AUTHOR: Tomoki Koike  
# DATE: OCT. 27 2019  
# DESCRIPTION: This program is designed to execute the class functions in the file  
# employee\_classes.py and execute each of the functions to make sure  
# that they are working fine.  
###  
  
# Importing the module  
**import** employee\_classes **as** empC  
  
**def main():** # User inputs  
 # Production worker  
 print**('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-')** employeeName **=** input**('Please enter the name of the employee -> ')** employeeNum **=** input**('Please enter the number of the employee -> ')** shiftNum **=** input**('Please enter the shift number for the production worker -> ')** hrPayRate **=** input**('Please enter the hour pay rate for this production worker -> ')** # Creating the object  
 print**()** prodWorkerData **=** empC.ProductionWorker**(**shiftNum, hrPayRate, employeeName, employeeNum**)** # Mutator and Accessor methods called  
 prodWorkerData.set\_EmployeeName**(**employeeName**)** print**(**prodWorkerData.get\_EmployeeName**())** prodWorkerData.set\_EmployeeNum**(**employeeNum**)** print**(**prodWorkerData.get\_EmployeeNum**())** prodWorkerData.set\_shiftNum**(**shiftNum**)** print**(**prodWorkerData.get\_shiftNum**())** prodWorkerData.set\_hr\_pay\_rate**(**hrPayRate**)** print**(**prodWorkerData.get\_hr\_pay\_rate**())** print**()** # print  
 line1 **= 'Employee Name: {0} >> Employee Number: {1}'** line2 **= '-------> Production worker'** line3 **= 'Shift #: {0} \n Hour pay rate: {1}'** print**(**line1.format**(**employeeName, employeeNum**) + '\n' +** line2 **+ '\n' +** line3.format**(**shiftNum, hrPayRate**))** # Shift Supervisor  
 # user inputs  
 print**('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-')** employeeName **=** input**('Please enter the name of the employee -> ')** employeeNum **=** input**('Please enter the number of the employee -> ')** yearSal **=** input**('Please enter the annual salary of the shift supervisor -> ')** yearBonus **=** input**('Please enter the annual bonus of this shift supervisor -> ')** # Creating the object  
 print**()** shiftAdvisorData **=** empC.ShiftSupervisor**(**yearSal, yearBonus, employeeName, employeeNum**)** # Mutator and Accessor methods called  
 shiftAdvisorData.set\_EmployeeName**(**employeeName**)** print**(**shiftAdvisorData.get\_EmployeeName**())** shiftAdvisorData.set\_EmployeeNum**(**employeeNum**)** print**(**shiftAdvisorData.get\_EmployeeNum**())** shiftAdvisorData.set\_yearSal**(**yearSal**)** print**(**shiftAdvisorData.get\_yearSal**())** shiftAdvisorData.set\_yearBonus**(**yearBonus**)** print**(**shiftAdvisorData.get\_yearBonus**())** print**()** # Printing results  
 line1 **= 'Employee Name: {0} >> Employee Number: {1}'** line2 **= '-------> Shift Supervisor'** line3 **= 'Annual Salary: {0} \n Annual Bonus: {1}'** print**(**line1.format**(**employeeName, employeeNum**) + '\n' +** line2 **+ '\n' +** line3.format**(**yearSal, yearBonus**))  
  
if** \_\_name\_\_ **== '\_\_main\_\_':** main**()**

**>>OUTPUT**

