1. Create an *Product* class.
   A *product* has

   - A name
   - A price
   - A quantity

   This class “looks” like

Product
name price quantity

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

2. Create a *Book* class.
   A *Book* has

   - title
   - author
   - page_count

   This class “looks” like

Book
title author page_count

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

3. Create a *Movie* class.
   A *Movie* has

   - title
   - director
   - runtime_minutes

   This class “looks” like

Movie
title director runtime_minutes

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

4. Create a *Song* class.

   A *Song* has

   - title
   - artist
   - duration_seconds

   This class “looks” like

Song
title artist duration_seconds

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

5. Create an *Employee* class.
   An *Employee* has

   - A name
   - A title
   - A salary

   An *Employee* can do

   - a greeting
   - request raise

   This class “looks” like

Employee
name title salary
greeting request_raise

   You should write getters and setters for each of the instance variables.

   A greeting should be of the form: Hello. My name is *name.* I'm the *title.*
   eg. Hello. My name is Eugene. I'm the CEO.

   A raise request should request a 6% raise.
   It should be of the form: I'm currently making *salary.* I'd like new salary of *new amount.*
   eg. I'm currently making $100. I'd like new salary of $106.

6. Create a *Student* class.
   A *Student* has

   - A name
   - A major
   - A GPA

   A *Student* can do

   - introduce themselves
   - study for exam

   This class “looks” like

Student
name major GPA
introduce study_for_exam

   You should write getters and setters for each of the instance variables.

   An introduction should be of the form: Hi, I'm *name.* I'm studying *major.*
   eg. Hi. I'm Maria. I'm studying Computer Science.

   Studying for an exam should increase the GPA by 0.2 points. (up to a maximum of 4.0)
   It should be of the form:
   I'm hitting the books! My GPA increased from *old GPA* to *new GPA*.
   eg. I'm hitting the books! My GPA increased from 3.5 to 3.7.

7. Create a *Vehicle* class.
   A *Vehicle* has

   - make
   - model
   - year

   A *Vehicle* can do

   - *print_vehicle_type*

   This class "looks" like

Vehicle
make model year
print_vehicle_type

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

   Write a method called *print_vehicle_type*, which prints in the form "[year] [make] [model]" example. "2021 Toyota Camry".

8. Create a *Course* class.
   A *Course* has

   - course_code
   - course_name
   - instructor

   An *Course* can do

   - *print_info*

   This class "looks" like

Course
course_code course_name instructor
print_info

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

   Write a method called *print_info*, which prints in the form
   "[course_code]: [course_name] taught by [instructor]"
   example. "CIS101: Introduction to programming taught by Matt".

9. Create a *Point* class.
   A *Point* has

   - x_coordinate
   - y_coordinate

   A *Point* can do

   - *print_info*

   This class "looks" like

Course
x_coordinate y_coordinate
print_info

   Create a constructor method that initializes all instance variables.
   You should write getters and setters for each of the instance variables.
   Instantiate an instance of the class. You may pass any initial values of your choosing.

   Write a method called *print_info*, which prints in the form
   "(x,y)=([x_coordinate], [y_coordinate])"

example. “(x,y)=(4, 5)”.

10. Create a *Vector* class.
A *Vector* has

    - x_direction
    - y_direction

    A *Vector* can do

    - get_magnitude

    Create a constructor method that initializes all instance variables.
You should write getters and setters for each of the instance variables.
Instantiate an instance of the class. You may pass any initial values of your choosing.

    Hint: magnitude is calculated as $\sqrt{x^2 + y^2}$.

    This class “looks” like

Vector
x_direction y_direction
get_magnitude

11. Create a *ColorRGB* class.
A *ColorRGB* has

    - red
    - green
    - blue

    A *ColorRGB* can do

    - to_grayscale

    Create a constructor method that initializes all instance variables.
You should write getters and setters for each of the instance variables.
Instantiate an instance of the class. You may pass any initial values of your choosing.

    The to_grayscale() method should return the grayscale value calculated as:

    $$0.3 * \text{red} + 0.59 * \text{green} + 0.11 * \text{blue}$$

    That is, it will just return a number (a float).

    This class “looks” like

ColorRGB
red green blue
to_grayscale

12. Create a *TemperatureInCelsius* class.

    A *TemperatureInCelsius* has

    - temp_value

    A *TemperatureInCelsius* can do

    - to_fahrenheit

    Clarification: temp_value is the temperature in Celsius.
Create a constructor method that initializes all instance variables.
You should write getters and setters for each of the instance variables.
Instantiate an instance of the class. You may pass any initial values of your choosing.

    The to_fahrenheit() method should return the temperature in Fahrenheit calculated as:
Fahrenheit = (Celsius * 9/5) + 32.

    This class “looks” like

TemperatureInCelsius
temp_value
to_fahrenheit

13. Create a *Rectangle* class.
    A *Rectangle* has

    - width
    - height

    A *Rectangle* can do

    - calculate_area

    This class “looks” like

Rectangle
width height
calculate_area

    Create a constructor method that initializes all instance variables.
    You should write getters and setters for each of the instance variables.
    Instantiate an instance of the class. You may pass any initial values of your choosing.

    The calculate_area() method should return the area calculated as: width * height.

14. Create a *Circle* class.
    A *Circle* has

    - radius

    A *Circle* can do

    - calculate_circumference

    This class “looks” like

Circle
radius
calculate_circumference

    Create a constructor method that initializes all instance variables.
    You should write getters and setters for each of the instance variables.
    Instantiate an instance of the class. You may pass any initial values of your choosing.

    The calculate_circumference() method should return the circumference calculated as: $2 \cdot \pi \cdot \text{radius}$.

15. Create a *Recipe* class.
    A *Recipe* has

    - name
    - cooking_time

    A *Recipe* can do

    - is_quick_meal

    This class “looks” like

Circle
name cooking_time
is_quick_meal

    Create a constructor method that initializes all instance variables.
    You should write getters and setters for each of the instance variables.
    Instantiate an instance of the class. You may pass any initial values of your choosing.

    The is_quick_meal() method should return True if the cooking_time is less than 30 minutes and False if it takes 30 minutes or more.