Lab 6 – Classes/Objects

SUBMIT original code in Python to solve the problem below.

Please be aware that copying and pasting code from any other source other than code you have explicitly written on your own is considered plagiarism. If you receive help, that is fine (document help in the comments of your code) however you need to write your own code, name your own variables, and comment your own code. Students turning in the exact same work as another student will all be given zeros. Plagiarism is not tolerated, and students found to be plagiarizing will be given a zero and reported to the University with the possibility of termination of the class and degree program.

Note: Lab six will be used and expanded in labs 7 and 10. If you do not complete lab 6, you will not be able to complete labs 7 and 10.

Write a class called **Worker** that encapsulates data about an office worker. The class should store the following things:

- Employee Number
- Office Number
- Name (First and Last)
- Birthdate
- Total number of hours worked
- Total number of overtime hours worked

Your class should also implement the following methods:

- **__init__()**
  - This is your constructor. It should not accept any arguments.
- **get_employee_number()**
  - Returns the employee number
- **set_employee_number(x)**
  - Changes the employee number
- **get_office_number()**
  - Returns the office number
- **set_office_number(x)**
  - If the office number given is less than 100 or greater than 500 return False, otherwise return True
- **get_name()**
  - Returns the employee's name
- **set_name(x)**
  - Changes the employee's name

- **set_birthdate(d, m, y)**
    - Changes the employee’s birthdate
    - The function should return true if the month is 1-12 and the day is 1-31 (don’t worry about which month has how many days) and false if an invalid day or month is entered.
- **get_hours_worked()**
    - Returns the number of hours worked
- **add_hours(x)**
    - If the number of hours being added is greater than 9, add 9 hours to the total hours worked and add the remainder to the total overtime
- **get_hours_overtime()**
    - Returns the number of overtime hours worked

You do not need to round or truncate your output. Name your class Worker and save your program as Lab6.py and upload just your class file to the appropriate dropbox in GradeScope, ***NOT D2L!***

REMEMBER

- Include the comment heading at the top of your code.
    - # Program Name: Lab1.py (use the name the program is saved as)
    - # Course: IT1114/Section XXX
    - # Student Name: John Doe
    - # Assignment Number: Lab#
    - # Due Date: xx/xx/ 20XX
    - # Purpose: What does the program do (in a few sentences)?
    - # List specific resources used to complete the assignment.

- Place comments within your code explaining the programming segments