

CMPS 1500 Introduction to Computer Science I – Spring 2019

Lab 1

Due **Wednesday 1/30/19** at 11:59 p.m. on Zybook and Canvas

You are required to work in a pair.

Lab guidelines: Complete the problems below. Submit the programming problems to Zybook for grading, and the text response to Canvas. Each program should be placed in its own file, named exactly as specified.

Your submission must be your work, and only your work. All program files should contain header docstrings with name(s) of program author(s), creation date, and a brief explanation of the program, as well as in-line comments explaining what your code does.

0. Seconds converter, `lab1pr0.py`

Write a program that asks the user for a number of seconds and tells them how many hours, minutes and seconds it is. Make sure your program works correctly for values smaller than the whole hour or whole minute. Here's what the sample output from several runs of your program would look like:

```
Please enter the number of seconds: 2
2 seconds
>>>
Please enter the number of seconds: 500
8 minutes 20 seconds
>>>
Please enter the number of seconds: 20000
5 hours 33 minutes 20 seconds
```

Hint: You may find `mod` and `div` operations helpful here. You will need to use `if` statements in this program.

1. SSN formatting, `lab1pr1.py`

In the United States, a Social Security number (SSN) is a nine-digit number that is a national identification number for taxation purposes. Write a program that asks the user for their social security number in `DDDDDDDDD` format (where `D` is digit). The program first performs a basic check if the number corresponds to an actual SSN by making sure it has 9 characters in it. If yes, it creates a string variable `new_ssn` with the SSN in the user-friendly `DDD – DD – DDDD` format and prints it. If not, it prints an error message `Incorrect SSN length`.

2. Date formatting, `lab1pr2.py`

Write a program that asks the user to enter a date in `MM/DD/YYYY` format that is typical for the US, the Philippines, Palau, Canada, and Micronesia. For the convenience of users from other nations, the program computes and displays this date in an alternative `DD.MM.YYYY` format.

A sample run of your program would look like this:

Please enter date in MM/DD/YYYY format: 7/4/1776

Here is the formatted date: 04.07.1776

You can assume that the user will enter a correctly formatted date, but do not count on having 0s in front of one-digit dates.

Hint: you will need to use string addition (concatenation) here.

3. More date formatting, lab1pr3.py

Write a program that asks the user to enter a date in MM/DD/YYYY format, typical for the US and Federated States of Micronesia. For the convenience of users from other nations, the program computes and displays this date in an alternative Month DD, YYYY format.

Please enter date in MM/DD/YYYY format: 07/04/1776

Here is the formatted date: July 04, 1776

Here you can assume that the user will enter a correctly formatted date but do not count on having 0s in front of one-digit dates.

Hint: You will need to create and use a list of month names in this program. In order to look up the name of a month, first get the month number and use it as an index in the list.

4. Last problem: Unity in diversity, lab1pr4.txt

For your answer you are welcome to consult any source you find credible, including internet sites and forums, family members, friends outside of class, your class colleagues, etc. Of course you may complete the answers on your own or with your partner. There is no one right answer. Watch the talk by Computerphile “The Problem with Time and Timezones” (10 minutes):

<https://www.youtube.com/watch?v=-5wpm-ges0Y>

As you can see in the talk, the diversity of users of a product will affect the way you will have to design the product. Give example situations when differences (e.g., in age, sex, gender identity, disability status, national origin, geographical location, cultural identity, economic status, language spoken, etc.) among the users of software applications and hardware devices affect the way we should design these applications and devices. Give at least four example scenarios (max is 7) and explain how the differences affect the design in 2-3 sentences for each case.