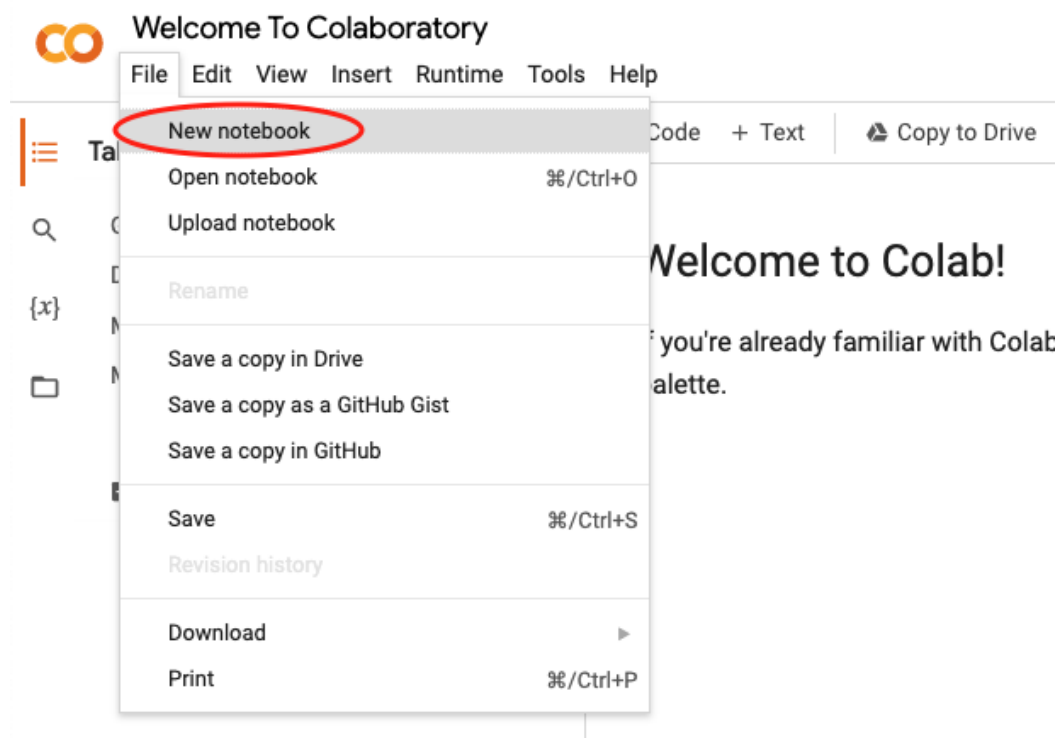


Lab 1

Introduction to Python Programming

I. Instruction:

1. Go to <https://colab.research.google.com/> , create a new notebook **Lab1_StudentID**.



2. Write all your code and answers in the Colab notebook. Create separate code sections for each Task.

The screenshot shows a JupyterLab interface with a notebook titled "Lab1_StudentID.ipynb". The top menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and "All changes saved". On the left sidebar, there are icons for a list, search, and a file explorer. The main area contains two cells: a text cell with "[]" and a code cell with the code `print("Hello class")` and its output "Hello class". Red arrows point to the "+ Code" and "+ Text" buttons at the top of the notebook, with labels "add code section" and "add text/note/answer" respectively.

3. Download the **Lab1_StudentID.ipynb** and submit on Blackboard

The screenshot shows a JupyterLab interface with a notebook titled "Ch3.ipynb". The top menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and "All changes saved". The "File" menu is open, showing options like "Locate in Drive", "Open in playground mode", "New notebook", "Open notebook", "Upload notebook", "Rename", "Move", "Move to trash", "Save a copy in Drive", "Save a copy as a GitHub Gist", "Save a copy in GitHub", "Save", "Save and pin revision", "Revision history", "Download", and "Print". The "Download" option is highlighted, and a sub-menu is shown with "Download .ipynb" circled in red. The background shows a code cell with a traceback error.

II. Tasks:

1. Create the variables $x = 2$ and $y = 3$, then determine what each of the following statements displays:
 - a) `print('x =', x)`
 - b) `print('Value of', x, '+', x, 'is', (x + x))`
 - c) `print('x =')`
 - d) `print((x + y), '=', (y + x))`

2. The following code should calculate the square of a number:

```
number = input("Input a number between 1 and 10")
square = number * number
```

What's wrong with this code? Fix that code and write your code in a new code section.

3. Test each of the arithmetic operators `+`, `-`, `*`, `/`, `//` and `**` with -5, 0, 5, and 7.5 as the left operand and 2 as the right operand.
4. Typically 6 eggs fit in one box. Write a script to calculate the number of boxes a farmer needs to store 28 eggs. The script will also determine how many eggs are placed in the last uncompleted box, and how many additional eggs are needed to fill this last box.
5. Use if statements to determine whether an integer is odd or even using the **remainder operator**.
6. Starting with 200 bacteria, the growth in the number of bacteria after n hours is calculated as follows: $B = 200 \times 2^n$. Print the number of bacteria after 0, 5, 10, and 15 hours in table format as shown below. Use the **tab escape sequence** to achieve the two-column output.

Hour	Number of Bacteria
0	200
5	6400
10	204800
15	6553600

7. Reimplement the code in Task 6 to produce the following table with the numbers **right aligned** in each column

Hour	Number of Bacteria
0	200
5	6400
10	204800
15	6553600

Hint: read <https://realpython.com/python-f-strings/> and <https://www.geeksforgeeks.org/string-alignment-in-python-f-string/>

8. Each of a string's **characters** has an integer representation. The set of characters a computer uses together with the characters' integer representations is called that computer's **character set**. You can indicate a character value in a program by enclosing

that character in quotes, as in "A." To determine a character's integer value, call the built-in function `ord`. Tom and Jim want to play a game but are unable to choose who goes first. They decide that the person with the highest sum of the integer values of the characters of their name gets to play first. Write a script to determine who goes first.

```
In [1]: ord('A')  
Out[1]: 65
```

9. Write a script that inputs the grades (in integers) of 3 courses from the user. First display the average grade of the three courses, and then the names and the grades of the courses with the highest and the lowest grades.
10. Write a script that inputs a number of seconds from the user. Calculate the number of hours, minutes, and remaining seconds. Print them separated by " - ". For example, if the user types 3750 seconds as input, the script should print:

1 - 2 - 30

Assume that the user enters a number of seconds that is **higher than 3600**. Use both the floor division and the remainder operation to calculate the number of hours, minutes, and seconds.

11. Every year, if an employee receives a good job performance review, they get a raise of 3% on their wages. In turn, if they receive a suboptimal performance review, their wages are deducted by 3%. Consider an employee starting with an hourly wage of \$10. Calculate how much this employee is earning after 5 years of consistent good reviews and after 2 years of bad reviews. Use the following formula to calculate these wages:

$$w = o(1 + p)^n$$

where

w is the new hourly wage,

o is the original hourly wage,

p is the percentage increase or decrease, and

n is the number of years with an increase or decrease in hourly wage.

12. While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your doctors and trainers. According to the American Heart Association (AHA) (<http://bit.ly/AHATargetHeart-Rates>), the formula for calculating your maximum heart rate in beats per minute is 220 minus your age in years. Your target heart rate is 50–85% of your maximum heart rate. Write a script that prompts for and inputs the user's age and calculates and displays the user's maximum heart rate and the range of the user's target heart rate.
13. After a running competition, three runners want to determine who won based on the time they needed to complete the course. Write a script that inputs the time they took through

three different floating-point numbers (in seconds) from the user. Display the numbers in increasing order by using **if statements** only. Prove that your script works by running it on all six possible orderings of the numbers. Does your script work with duplicate numbers?

14. Reimplement code in Task 13 by using **if...else statements** rather than only **if statements**.