

# Introduction to programming - Demo 2

## General instructions

- Questions are worth 10 points and final question is 20 points.
- Put each task in **separate file** and name them so you find them easily, like demo1\_1.py and demo1\_4.py
- Use IDLE or other program to type and **TEST** your programs. Test thoroughly to make sure program works with any (reasonable) input
- Bring in your solutions to next demonstration session in flash drive or save them in a web folder accessible in class.
- Remember to comment your code - this does not mean, that every single line should be commented. Comment the important parts in your program.
- **The code does not matter:** The purpose of programming is to solve the problem. How you exactly program it does not matter. Demonstrator has often shorter or more efficient solution, but that comes with programming experience. Do not consider your solution wrong, even if it looks different. We'll always discuss multiple solutions for each task and figure out which might be the "best". That is often subjective.
- Points are awarded for working solutions, but also for *trying*. Please mark the tasks that you are willing to present in class. Even if the program does not work yet, we can figure it out together.
- In case you want to do "extra" work and extend the programs further, take a copy of your solution and extend that. If it's your turn to present a solution, show the minimum solution first and show off any additional work after that.
- Example solutions will be published later.

## Preface

This weeks tasks consist primarily of reading user input, and most tasks are also simple math problems. If math is a turn-off for you, remember that math is always in the background in programming, even if you create some non-math programs. The point is that we just write the formula for the computer and let it do the work for us.

## Task 1.

Start with the following program:

```
a = 3
b = 6
c = 7
```

Extend the program by adding two new variables called **total\_sum** and **average**. Calculate the sum of **a**, **b** and **c** and assign the value to **total\_sum**. Calculate the average of the numbers and assign the value to variable **average**.

Finally, output both **total\_sum** and **average**.

---

## Task 2.

Write a program that queries the user for a name, and outputs a message "Hello there, name"

Example execution:

```
What is your name? John Smith
Hello there, John Smith
```

---

## Task 3.

Write a program that queries the user for a temperature in Celsius, and then outputs the temperature in Fahrenheit. The formula for converting temperature to Fahrenheit is

$$\frac{9}{5} \cdot temp\_in\_celsius + 32$$

Example execution:

```
Give a temperature in Celsius: 25
25 degrees Celsius is 77.0 degrees Fahrenheit.
```

## Task 4.

Write a program that queries users name and year of birth, and then outputs the string displaying users name and age in 2020 (within accuracy of one year)

```
Give your name: John Smith
Which year you were born: 1981
Hello, John Smith. You are 39 years in 2020
```

Extra discussion: Can you make it to work in any year?

---

## Task 5.

Write a program that queries a string and an integer, and then outputs the string repeated the given number of times

Example execution:

```
Give a string: abc
Give a number: 3
abccabccabc
```

(Hint: this does not need a loop which you might want to use if you are familiar with other languages)

---

## Task 6. (Double points!)

Write a program that queries user for three integers: **a**, **b** and **c** and solves the quadratic equation  $ax^2 + bx + c = 0$

The roots can be solved by the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

In order to calculate square roots in python, you need to use function `sqrt(x)`. To use the function, put following as the first line of your program:

```
from math import sqrt
```

Example program that is using the squareroot function:

```
from math import sqrt
my_number = 49
root = sqrt(my_number) # square root of 49 is 7
print(root)
```

**NOTE!** Your program doesn't need to solve equations with imaginary roots; hence make sure the program works with (at least) following inputs:

```
a: 1
b: 2
c: -8
roots:
2.0
-4.0
```

```
a: 2
b: 5
c: 3
roots:
-1.0
-1.5
```

```
a: 1
b: -3
c: 0
roots:
3.0
0.0
```

---