

Computational Maths

**CT4032**

**Workbook – Week 1**

2023 – 24

**Dr Nasreen Anjum**

School of Computing and Engineering

**University of Gloucestershire 2023/24**

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, including – but not limited to – photocopy, recording, or any information storage and retrieval system, without the specific prior written permission of University of Gloucestershire.

1

Introduction to Python Calculator

Week 1 – Python basics and calculations

**Introduction to Python**

This work sheet will cover some of the basic statements and defining functions with python. Check to see if it downloaded by going to the start menu and checking IDLE. You don’t need the newest version (currently 3.10.7).

A screen shot of a computer

Description automatically generated

If not, use the this link <https://www.python.org/downloads/>. We will be using python for most of the practical session and going through how we can convert the discussed theory into computer code. Python is more of a scripting language, so having different files for the practical sessions or any other pieces of code that you want to have is fine.

# **First Commands**

The first statement we will cover is “print”, which in this case will output any given string. This will display it as “Hello World!” and anything else that is within the brackets.

**--- Code ---**

print(“Hello World!")

**--- Output ---**

Hello World!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Variables**

We can now use another statement to designate variables and give them values which can be used to then display. You can also print strings and variables you state together. These variables can be setup before the start of a program, or can have values assigned to them during the program’s runtime.

**--- Code ---**

v1 = 5

print(v1)

print(“There are “, v1, “ Apples”)

**--- Output ---**

There are 5 Apples

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Calculations**

We can now expand upon the two previous statements together and perform some basic calculations. Below is the example for addition. First, we define a function which we can use to perform addition without adding additional lines. Afterwards we use | v1 = int(input("Enter the first number: ")) | specifically “int” to designate the input variable as an integer.

**--- Code ---**

def addition(x, y):

return x + y

v1 = int(input("Enter the first number: "))

v2 = int(input("Enter the second number: "))

print( v1, ' + ', v2, ' = ', addition(v1, v2))

--- Output ---

Enter the first number: << (input 5)

Enter the second number: << (input 8)]

5 + 8 = 13

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# If else

Usage of if else statements can be used to add conditions to variables or inputs which then provide more paths for your code. The simple if statement below is to compare two numbers and check which one is greater.

--- Code ---

v1 = int(input("Input the first number: "))

v2 = int(input("Input the second number: "))

if v1 > v2:

print("first is greater than the second")

else:

print("second is greater than the first")

**--- Output ---**

Input the first number: << 8

Input the second number: << 5

first is greater than the second

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Loops**

Python can loop, there are two different times that will be discussed here. Firstly, is this the “for” loop. For the first one its just a simple from 0 to 10. The second is starting on 3 and ending on 9. The last one is starting on 1 and ending at 15, but going up in increments of 2.

**--- Code ---**

For c1 in range(10):

Print(c1)

For c2 in range(3, 9):

Print(c2)

For c3 in range (1, 15, 2):

Print(c3)

**--- Output --- (Going downwards in python, run the code to see)**

0 1 2 3 4 5 6 7 8 9

3 4 5 6 7 8

1 3 5 7 9 11 13

**Python on Arrays**

Arrays of a group of variables stored under a single list. This can be manipulated by splitting, adding, or removing variables from the list. This can be a useful method of storing your equation and applying or performing functions in a way to calculate it. The command “len()” is used to generate the number of items in the array.

**--- Code ---**

parray = ['13', '+', '15']

print(parray)

print(parray[0])

num = len(parray)

print(num)

**--- Output ---**

[“13”, “+”, “15”]

13

3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Coding Challenges

1. **Create a Simple Calculator**

Objective: Create a Python program that allows the user to perform basic arithmetic operations: addition, subtraction, and multiplication.

**Instructions:**

1. Begin by displaying a welcome message to the user, explaining what the program does.
2. Ask the user to input two numbers (you can use input() for this).
3. Present the user with a menu to choose an operation:

* Press '1' for addition.
* Press '2' for subtraction.
* Press '3' for multiplication.

1. Ask the user to enter their choice.
2. Based on the user's choice, perform the corresponding operation on the two numbers and display the result.
3. Give the user the option to perform another calculation or exit the program.
4. If the user chooses to perform another calculation, loop back to step 2. If they choose to exit, display a thank you message and end the program.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Check if a Number is Prime**

**Objective**: Create a Python program that checks whether a given number is prime or not.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a positive integer.
3. Check if the entered number is prime or not. A prime number is a positive integer greater than 1 that has no positive integer divisors other than 1 and itself.
4. Display the result, indicating whether the number is prime or not.
5. Ask the user if they want to check another number or exit the program.
6. If the user chooses to check another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.