



Columbia College
Vancouver, Canada

**Introduction to Computer Science and Programming 1
CSCI120**

Chapter4: Functions

Assignment 4

Note: This document has been designed and developed as part of an initiative for creating an OER (Open Education Resource) package for the course CSCI 120 at Columbia College.

Please contact Alireza.davoodi@gmail.com for any comment, modification, and questions.

Terms of use: Please feel free to customize this document as needed

Last Modified: May 2022



Requirements

- Please use meaningful name for your variables and functions
- Try to reuse your solutions as much as possible.
- For each of the following problem you need to
 - o Define a function (for instance function1)
 - o Add comments for the functions as discussed in the lecture
 - o Mention the list of input parameters
 - o Mention what type of output does the function has
 - o Write a test function for instance testFunction1
 - o In the test function, for all test cases you have already written for your algorithm, write a function call inside the main function
 - o Call the test function in the main function.
- Define all the functions in one file (all in one)
- Define the function header
- Define a main function
- Call the functions inside the main function
- If the function you implement for a question is big, please try to break down to multiple functions.
- Do not use methods, functions, statements that we have not covered in the previous lectures.

##Problem 1

```
print("Problem1-----")
```

Python code for problem 1

##Problem 2

```
print("Problem2-----")
```

Python code for problem 2

If it is a group assignment, please add the information here

# of Students in the Group:		
Student 1	First name, last name	Student-ID
Student 2	First name, last name	Student-ID
Student 3	First name, last name	Student-ID
Student 4	First name, last name	Student-ID



Problem1

- Design and implement a function with an input parameter which is a positive number and prints and returns the sum of the number's digits. For instance if the number is 123 the algorithm returns 6 which is the result of $1+2+3$.

Problem2

- Design and implement a function with one parameter which is an integer and finds the next prime number which is bigger than the given input parameter and returns it.
- Suggestion: Define and implement a function called isPrime which checks whether a number is prime or not and then reuse it in this problem.

Problem3

- Design and implement a function with two input parameters, A and B. The functions then calculates the result of the floor division of A over B ($A//B$). You are not allowed to use the floor division operator. Look at here: [https://simple.wikipedia.org/wiki/Division_\(mathematics\)](https://simple.wikipedia.org/wiki/Division_(mathematics))
- For instance the function for 20 and 6 will return 3.

Problem4

- Design and implement a function with no input parameter which reads a number from input (like 123). Only non-decimal numbers are valid (floating points are not valid). The number entered by the user should not be divisible by 10 and if the user enters a number that is divisible by 10 (like 560), it is considered invalid and the application should keep asking until the user enters a valid input. Once the user enters a valid input, the program calculates the reverse of the input number (for 153, the reverse is 351) and prints the result and returns the results.

Problem5

- Write a function called printSubLists which receives two number A and B as its parameters:
- First prints all numbers between A and B (A and B not included), which are divisible to both 3 and 5.
- Then prints all numbers between A and B (A included by B not included), which are divisible by either 6 or 7.



Columbia College

Vancouver, Canada

- Finally prints all numbers between A and B (A and B both included), which are not divisible by 3.
- Hint: Design a function for each sub problem and then call them inside the printSubLists function.

Good Luck ☺