

Table of Contents

[Part A (Programming Fundamentals) 3](#_Toc177748234)

[Task 1: Variables and Datatypes 3](#_Toc177748235)

[Task 2: Arithmetic Operators 4](#_Toc177748236)

[Task 3: Comparison Operators 5](#_Toc177748237)

[Task 4: Logical Operators 7](#_Toc177748238)

[PART B (Functions, and Data Structures) 9](#_Toc177748239)

[Task 5: Simple Function with Parameters 9](#_Toc177748240)

[Task 6: Function with Arrays 10](#_Toc177748241)

[Task 7: Data structures: List 12](#_Toc177748242)

[Task 8: Data structures: Dictionary 19](#_Toc177748243)

[References 24](#_Toc177748244)

# Part A (Programming Fundamentals)

## Task 1: Variables and Datatypes

Coding Of Calculating momentum

A screen shot of a computer code

Description automatically generated

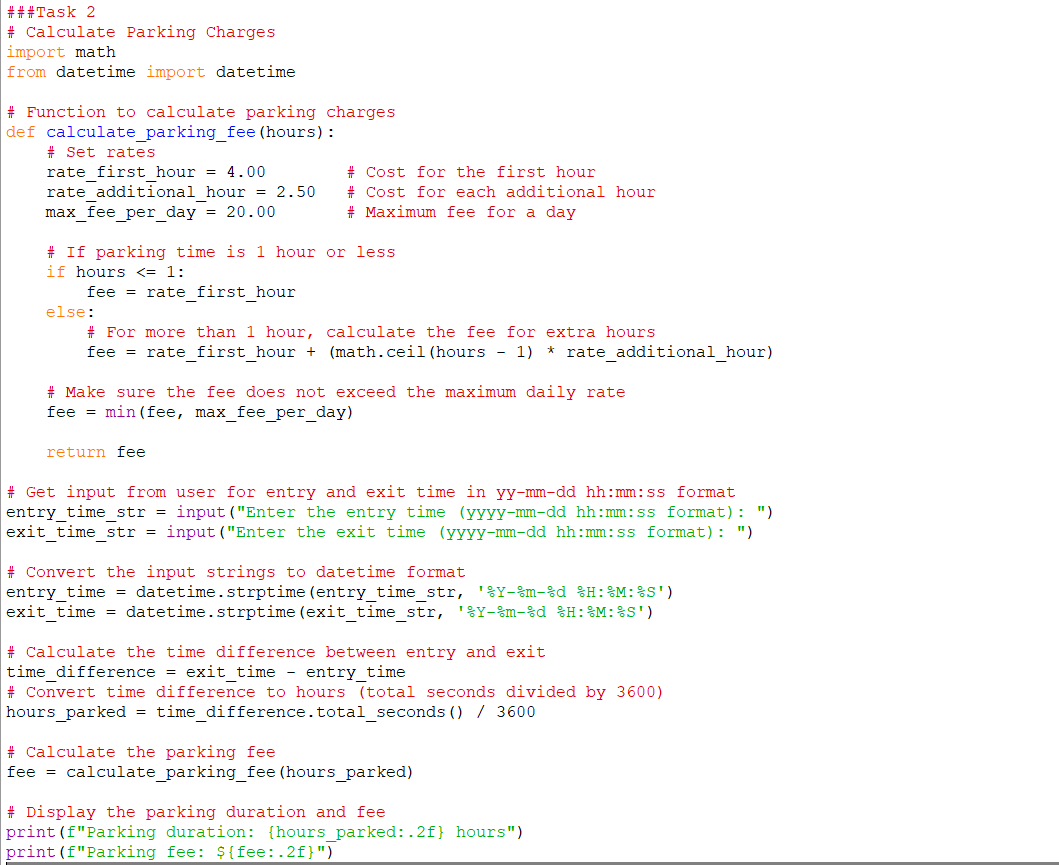
This code shows a momentum as a function, calculate\_momentum(mass, velocity), and the formula of momentum is p=mvp = mvp=mv where mass is in kilograms and velocity in meters per second. These values are then asked from the user and are converted to floating-point number and after that the momentum function is called. Last of all, it displays the computed momentum in kg⋅pm/s kg⋅m/s kg⋅pm/s.

Outcome of the coding



## Task 2: Arithmetic Operators

Coding parking lot calculation



This code refers to the counting of the parking fees that are determined by the difference in the time of entry of the car and the time of exit. The user inputs the entry and exit times in the format yyyy-mm-dd hh:mm:ss in which they are then converted to python’s datetime. This part tells the number of hours the car has been parked in total. For parking duration of one hour or less, there is a fixed rate charged subjected to the parking tariffs. Apart from that, they have an additional rate for any more hours in the program which cannot go beyond the daily maximum charge. Last but not the least, in the final line, we print the parking duration along with total fee needed to be paid.

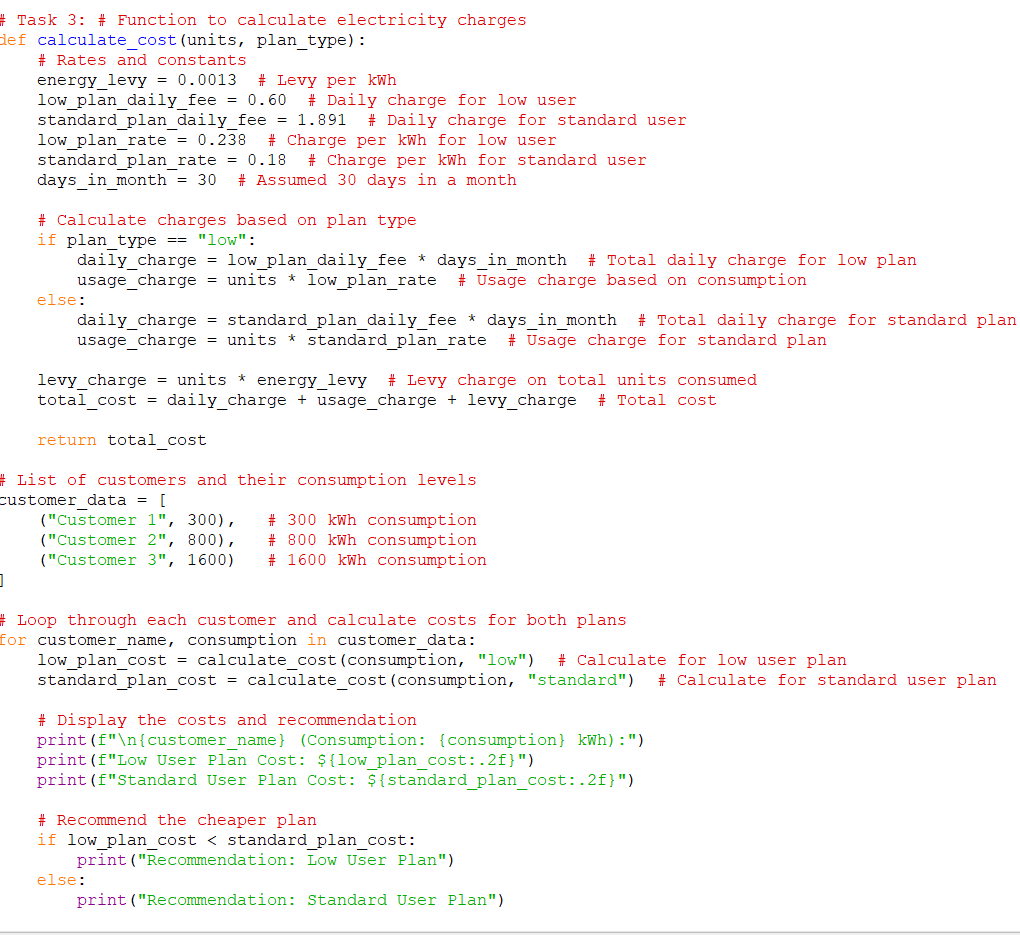
Outcome of the code

A close-up of a computer code

Description automatically generated

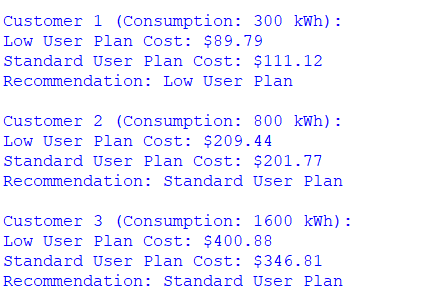
## Task 3: Comparison Operators

Electricity charges and most economical options for the customers based on their usage



This code is designed to generate electricity charges according to customers’ usage and customer type: low or standard user. It spells out charges per day, charges per kilowatt hour, and what it calls an energy tariff levy. The function includes the daily charge, the usage charge based on the consumption of goods, services, or products and the levy in the total cost. For each customer, the algorithm works through the low user plan and the standard user plan and outputs the costs of both the plans, also recommending the cheaper plan. It follows simple variable and simple logic structures to facilitate simple grasp by its users.

Otcome



## Task 4: Logical Operators

A screenshot of a computer program

Description automatically generated

This code determines the weekly payable for an employee where; The function calculate\_weekly\_pay first calculate the overtime as 1. This is paid five times the hourly wage so it will come in handy in case one would need to pay high fees. It then adds the basic wage for the hour worked then multiplies the hourly wages to give the regular pay in addition to the overtime rate to give the overtime pay. Total weekly pay means the pays that are got from the normal working hours plus the pay for the overtime services rendered The two different pays are then added together to give the total amount of weekly pay. The main one gets the input of the hourly wage, regular hours and overtime hours, checks whether the hours are not negative and then calls the function to calculate the total weekly pay.

Outcome of the code

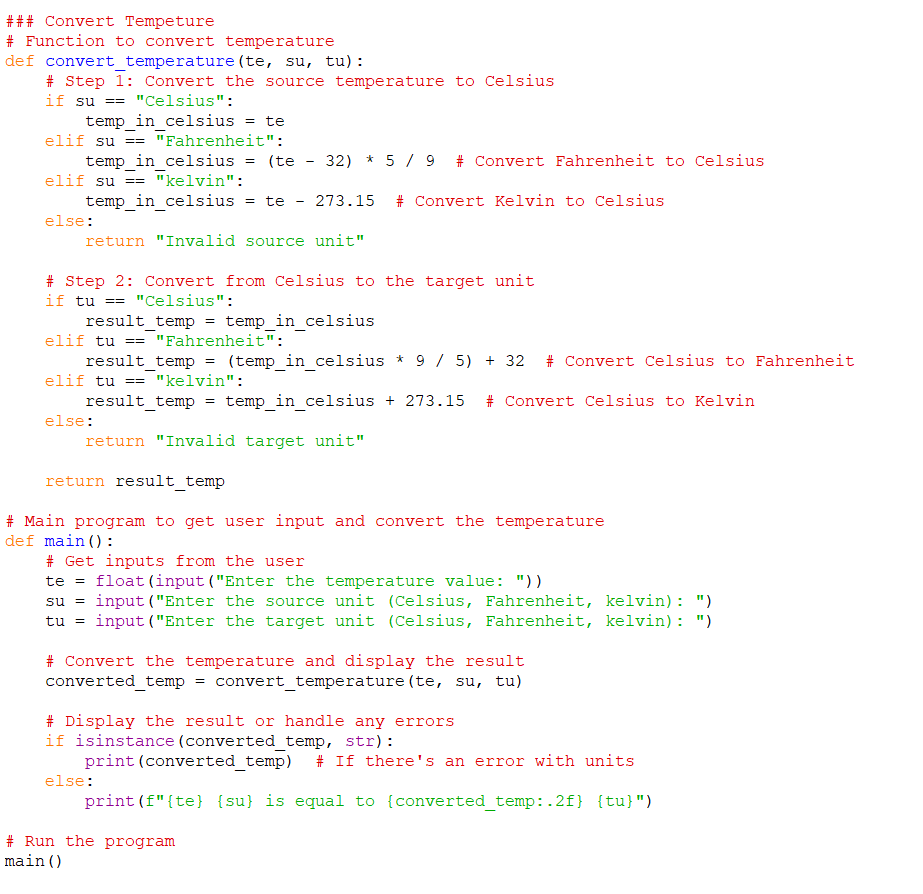
A white background with blue text

Description automatically generated

# PART B (Functions, and Data Structures)

Task 5: Simple Function with Parameters

Convert Temperature



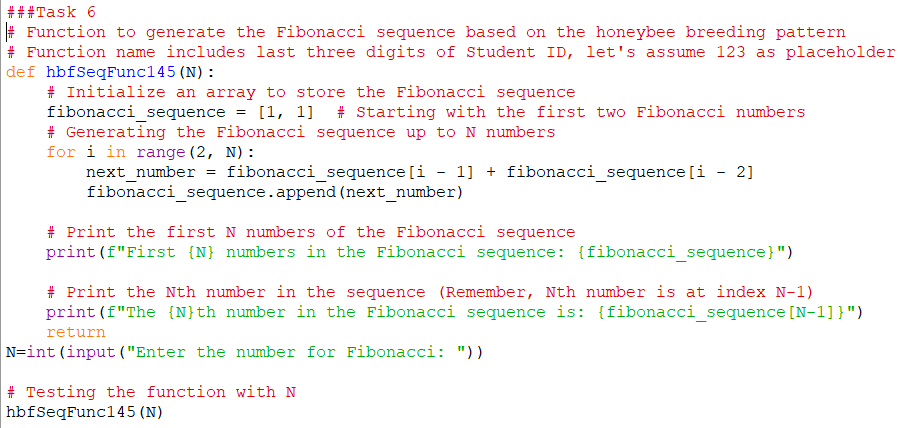
This is an algorithm that transforms a given temperature measurement from one system to another (Celcius, Fahrenheit or Kelvin). To be more precise, the function convert\_temperature at first the input temperature into Celsius based on the source unit given by the user. If the source unit is fahrenheit, or kelvin, the temperature is converted to Celsius using the standard formulae. Then it jumps ahead and evaluates the celsius value to the target unit which could be either fahrenheit or kelvin if the case may be. The first primary function prompts the user to enter the temperature value, source unit and target unit then converts the temperature and returns the result. If the user enters wrong units again, program gives the likely units respective error message.

Outcome of the code

A close-up of a number

Description automatically generated

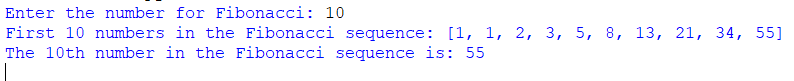
## Task 6: Function with Arrays

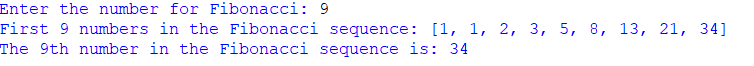
Fibonacci Sequence in Honeybees breeding

This code computes and prints out the Fibonacci series of numbers given a user defined number NNN. This function is named hbfSeqFunc145 and it starts list with 1, 1 and over using a loop, it go on adding other terms up to NNN, since N is assumed to take 3-digit integer values. The n+1 th number is the sum of the two numbers that come immediately before it and this is added on the list. Finally, the function creates the sequence of the Fibonacci numbers, then outputs first NNN numbers in the Fibonacci sequence and indicates the position of NNNth number – this position is calculated with the help of formula and is equal to N−1N-1N−1. The user is required to provide the value of NNN after which he or she is directed to the function to perform the computation.

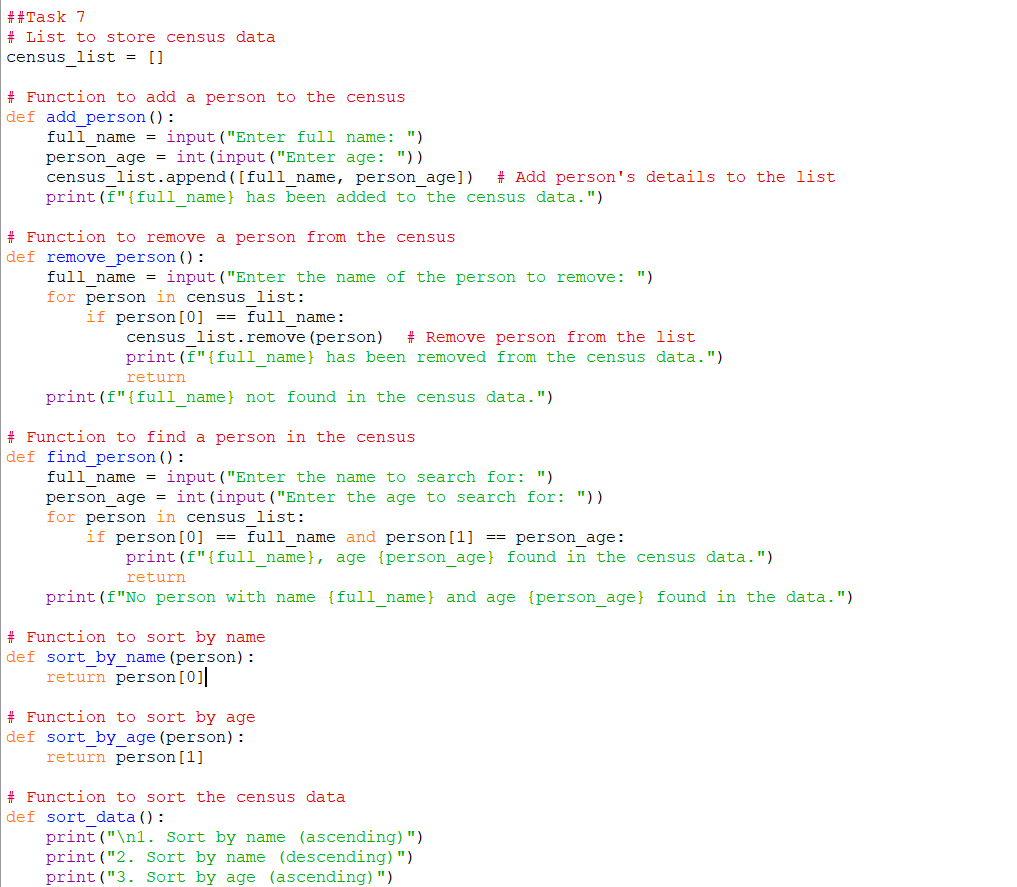
Outcome of the code







Task 7: Data structures: List



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Here is the code of one of the basic console applications that can be created to manage census data; it includes the users’ ability to insert, delete, search, sort and output records that contain names and ages of people. It involves the use of a list of dictionaries in storing and accessing each persons data hence easy to manipulate. Each action has corresponding functions such as adding record, searching, sorting by name or age of the people in the list. Is an iterative type of program where it will display the menu of options to the user and when a user decides to quit the program. Altogether, it can be considered as an easy-to-use tool in regards to the main processes of census administration.

Outcome of the code

Full name and age adding

A screenshot of a computer program

Description automatically generated

Searching for the person and found

A screenshot of a computer code

Description automatically generated

Removing data

A white background with blue text

Description automatically generated

Sorted data by age

A screenshot of a computer code

Description automatically generated

Sorted data by Full name

A screenshot of a computer code

Description automatically generated

Showing all the list and exiting the programme.

A screenshot of a computer program

Description automatically generated

Task 8: Data structures: Dictionary A screenshot of a computer code

Description automatically generated

A computer screen shot of a program

Description automatically generated

This code maintains simple game inventory and the inventory has been implemented using dictionary in which the item and associated quantity have been stored. It offers several functions to represent various parts of the inventory, to show all the items, to show the bullet count, or to list names or quantities of the items. In this case, the program is designed in bear bone form, the user can perform actions in the repetitive manner until he or she decides to stop. All the functions are well defined for a particular function and thus the code is very easy to understand.

Outcome of the code

Showing all quantities and items

A screenshot of a computer

Description automatically generated

Showing bullet quantitie

A white screen with blue text

Description automatically generated

Showing all itemes names

A white screen with blue text

Description automatically generated

Showing all item quantities

A screenshot of a computer

Description automatically generated

Exiting the programme

A white background with blue text

Description automatically generated

# References

W3Schools. (1998). W3Schools Online Web Tutorials. W3schools.com; W3Schools. https://www.w3schools.com/

‌