Challenges – Unit 1 (Operators)

1. Print three different strings.

2. Write a program that prints a message if a variable is less than 10, and different message
if the variable is greater than or equal to 10.

3. Write a program that prints a message if a variable is less than or equal to 10, another
message if the variable is greater than 10 but less than or equal to 25, and another message
if the variable is greater than 25.

4. Create a program that divides two variables and prints the remainder.

5. Create a program that takes two variables, divides them, and prints the quotient.

6. Write a program with a variable age assigned to an integer that prints different strings
depending on what integer age is.

7. Write a program that takes and input number from a user and then determines if that number is even or odd.

b. create a ‘try’ and ‘exception’ clause for if the user inputs zero as the denominator. If they enter zero the program should say ‘invalid input.’

8. Write a program that calculates the population of a country for the next four years, assuming it starts with an initial population of 10,000 in 2020, and the number of people increases by 10% each year.

Challenges Unit 1 – (Functions)

1. Write a function that takes a number as an input and returns that number squared.

2. Create a function that accepts a string as a parameter and prints it.

3. Write a function that takes three required parameters and two optional parameters.

4. Write a program with two functions. The first function should take an integer as a parameter
and return the result of the integer divided by 2. The second function should take an integer
as a parameter and return the result of the integer multiplied by 4. Call the first function, save
the result as a variable, and pass it as a parameter to the second function.

5. Write a function that converts a string to a float and returns the result. Use exception
handling to catch the exception that could occur.

6. Add a docstring to all of the functions you wrote in challenges 1–5.

Challenges Unit 1 – (Math)

1. Egg cartons each hold exactly 12 eggs. Write a program which reads an integer number of eggs from input, then prints out two numbers: how many cartons can be filled by these eggs, and how many eggs will be left over.

2. Write a program that reads two positive integers A and B on separate lines. If A is divisible by B, print the message “divisible.” Otherwise print the message “not divisible.”

3. The geometric mean of two number A and B is $\sqrt{ab}$ the number:

It is commonly used to compare aspect ratios of display screens and describe the average growth rate of a population. Write a program that reads two lines of positive float from input, and outputs their geometric mean.

4. A parcel is thrown downward at a speed of *v* m/s from an airplane at altitude 11000 m. As it falls, its distance from the ground is given by the formula -4.9*t*2 – *vt* + 11000, where *t* is the time in seconds since it was dropped. Write a program to output the time it will take to reach the ground. The input to your program is the positive floating-point number *v*. The required time is given by the quadratic formula:

$$t = \frac{v - \sqrt{v^2 - 4(-4.9)(11000)}}{2(-4.9)}$$

5. Write a program that converts one unit of measurement to another (ie. Degrees Fahrenheit, to degrees Celsius)

6. Write a program to calculate the volume and surface area of a cylinder given user input.

7. Write a program that finds and prints all the multiples of a number under 20.

8. Write a program that calculates and displays the *n* th term of the Fibonacci sequence. (This is tough, might need to use a ‘for’ loop – we haven’t learned it yet.

Challenges Unit 1 – Loops (Intro)

1. Use a for loop to display the numbers 0 to 10.

2. Ask the user to enter their name. Use a for loop to display their name 3 times.

3. Ask the user to input a starting number. Display that number and the next three numbers after that (i.e. if they enter 4 it should display 4, 5, 6 and 7).

4. Alter the previous program to ask the user for a starting number and an ending number. Display *all* the numbers in that range (including both the starting and ending numbers they entered).

5. Ask the user to input a number. If that number is more than 10 display the message “Too high” and ask them to try again and input another number. Keep repeating this loop until they enter a number which is less than or equal to 10.

6. Ask the user to input a number and then ask if they want to double the number. If they answer “y” multiply the number by 2 and display the answer. Keep repeating this loop, doubling their number each time, until they no longer reply “y”.

7. Ask the use if they want a cup of tea. If they reply with “no” or “n” repeat the question. Once the loop stops display the message “Sorry, we have run out of tea”.

Challenges Unit 1 – Containers

1. Create a list of your favorite musicians.

2. Create a list of tuples, with each tuple containing the longitude and latitude of somewhere
you've lived, visited, or wish to visit.

3. Create a dictionary that contains different attributes about you: height, favorite color,
favorite author, etc.

4. Write a program that lets the user ask your height, favorite color, or favorite author, and
returns the result from the dictionary you created in the previous challenge.

5. Create a dictionary mapping your favorite musicians to a list of your favorite songs by
them.

Challenges Unit 1 – Containers More Practice

1. Create a tuple containing the names of five countries and display the whole tuple. Ask the user to enter one of the countries that have been shown to them and then display the index number (ie. Position in the list) of that item in the tuple

2. Add to the previous program to ask the user to enter a number and display the country in that position

3. Create a list of two sports. Ask the user what their favourite sport is and add this to the end of the list. Sort the list and display it.

4. Create a list of six school subjects. Ask the user which of these subjects they don’t like. Delete the subject they have chosen from the list before you display the list again.

5. Ask the user to enter four of their favourite foods and store them in a dictionary so that they are indexed with numbers starting from 1. Display the dictionary in full, showing the index number and the item. Ask them which they want to get ride of and remove it from the list. Sort the remaining data and display the dictionary.

6. Enter a list of ten colours. Ask the user for a starting number between 0 and 4 and an end number between 5 and 9. Display the list for those colours between the start and end numbers the user input.

7. Create a list of four three-digit numbers. Display the list to the user, showing each item from the list on a separate line. Ask the user to enter a three digit number. If the number they have typed in matches one in the list, display the position of that number in the list, otherwise display the message ‘that is not in the list.’

8. Ask the user to enter the names of three people they want to invite to a party, and store them in a list. After they have entered all three names, ask them if they want to add another. If they do, allow them to add more names until they answer ‘no’. When they answer ‘no’, display how many people they have invited to the party.

Challenges Unit 1 - Strings

1. Print every character in the string "Camus".

2. Write a program that collects two strings from a user, inserts them into the string
"Yesterday I wrote a [response_one]. I sent it to [response_
two]!" and prints a new string.

3. Use a method to make the string "aldous Huxley was born in 1894." grammatically correct by capitalizing the first letter in the sentence.

4. Take the string "Where now? Who now? When now?" and call a method that
returns a list that looks like: ["Where now?", "Who now?", "When now?"].

5. Take the list ["The", "fox", "jumped", "over", "the", "fence",
"."] and turn it into a grammatically correct string. There should be a space between each
word, but no space between the word fence and the period that follows it. (Don't forget, you

learned a method that turns a list of strings into a single string.)

6. Replace every instance of "s" in "A screaming comes across the sky." with a dollar sign.

7. Use a method to find the first index of the character "m" in the string "Hemingway".

8. Find dialogue in your favorite book (containing quotes) and turn it into a string.

9. Create the string "three three three" using concatenation, and then again using
multiplication.

10. Slice the string "It was a bright cold day in April, and the clocks were striking thirteen." to only include the characters before the comma.

Challenges Unit 1 – Loops Cont'd

1. Print each item in the following list: ["The Walking Dead", "Entourage", "The Sopranos", "The Vampire Diaries"].

2. Print all the numbers from 25 to 50.

3. Print each item in the list from the first challenge and their indexes.

4. Write a program with an infinite loop (with the option to type q to quit) and a list of
numbers. Each time through the loop ask the user to guess a number on the list and tell them
whether or not they guessed correctly.

5. Multiply all the numbers in the list [8, 19, 148, 4] with all the numbers in the list
[9, 1, 33, 83], and append each result to a third list.

Challenges Unit 1 – Random

1. Randomly Choose a number between 1 and 5. Ask the user to pick a number. If they guess correctly, display the message 'Well Done.' Otherwise tell them if they are too high or too low and ask them to pick a second number. If they guess correctly on their second guess display 'Correct', otherwise display 'you lose.'

2. Randomly pick a whole number between 1 and 10. Ask the user to enter a number and keep entering numbers until they enter the number that was randomly picked.

3. Update the last program (#2) so that it tells the user if they are too high or too low before they pick again.

4. Make a math quiz that asks five questions by randomly generating two whole numbers to make an equation (eg. [num1] + [num2]). Ask the user to enter the answer. If they get it right add a point to their score. At the end of the quiz tell them how many they got correct out of 5.

Challenges Unit 1 – Arrays

1. Ask the user for a list of five integers. Store them in an array. Sort the list and display it in reverse order.

2. Create an array which will store a list of integers. Generate five random numbers and store them in the array. Display the array (showing each item on a separate line).

3. Ask the user to enter numbers. If they enter a number between 10 and 20, save it in the array, otherwise display the message “Outside the range”. Once five numbers have been successfully added, display the message “Thank you” and display the array with each item shown on a separate line.

4. Create an array which contains five numbers (two of which should be repeated). Display the whole array to the user. Ask the user to enter one of the numbers from the array and then display a message saying how many times that number appears in the list.

5. Create two arrays (one containing three numbers that the user enters and one containing a set of five random numbers). Join these two arrays together into one large array. Sort this large array and display it so that each number appears on a separate line.

6. Ask the user to enter five numbers. Sort them into order and present them to the user. Ask them to select one of the numbers. Remove it from the original array and save it in a new array.

7. Display an array of five numbers. Ask the user to select one of the numbers. Once they have selected a number, display the position of that item in the array. If they enter something that is not in the array, ask them to try again until they select a relevant item.

8. Create an array of five number between 10 and 100 which each have two decimal places. Ask the user to enter a whole number between 2 and 5. If they enter something outside of that range, display a suitable error message and ask them to try again until they enter a valid amount. Divide each of the numbers in the array by the number the user entered and display the answers shown to two decimal places.

Challenges Unit 1 – 2D Lists and Dictionaries

1. Create the following using a simple 2D list using the standard Python indexing:

	0	1	2
0	2	5	8
1	3	7	4
2	1	6	9
3	4	2	0

2. Using the 2D list from the previous program, ask the user to select a row and a column and display that value.

3. Using the 2D list from question 1, ask the user which row they would like displayed and display just that row. Ask them to enter a new value and add it to the end of the row and display the row again.

4. Change your previous program to ask the user which row they want displayed. Display that row. Ask which column in that row they want displayed and display the value that is held there. Ask the user if they want to change the value. If they do, ask for a new value and change the data. Finally, display the whole row again.

5. Create the following using a 2D dictionary showing the sales each person has made in the different geographical regions:

	N	S	E	W
John	3056	8463	8441	2694
Tom	4832	6786	4737	3612
Anne	5239	4802	5820	1859
Fiona	3904	3645	8821	2451

6. Using the program from Question 5, ask the user for a name and a region. Display the relevant data. Ask the user for the name and region of data they want to change and allow them to make the alteration to the sales figure. Display the sales for all regions for the name they choose.

7. Ask the user to enter the name, age and shoe size for four people. Ask for the name of one of the people in the list and display their age and shoe size.

8. Adapt program in the previous example to display the names and ages of all the people in the list but do not show their shoe size.

9. After gathering the four names, ages and shoe sizes, ask the user to enter the name of the person they want to remove from the list. Delete this row from the data and display the other rows on separate lines.