Name:	Owen	Course Code:	

Workbook

Contents

Course Introduction 4
What is a programmer? 7
What is Python? 9
Set up your development environment 11
More About Programs 12
Debugging 13
SoloLearn – Basic Concepts – What is Python? 14
SoloLearn – Basic Concepts – Your First Program 15
SoloLearn – Basic Concepts – Simple Operations 16
SoloLearn – Basic Concepts – Floats 17
SoloLearn – Basic Concepts – Other Numerical Operations 18
SoloLearn – Basic Concepts – Strings 19
SoloLearn – Basic Concepts – Simple Input & Output 20
SoloLearn – Basic Concepts – String Operations 21
SoloLearn – Basic Concepts – Type Conversion 22
SoloLearn – Basic Concepts – Variables 23
SoloLearn – Basic Concepts – In-Place Operators 25
SoloLearn – Basic Concepts – Using an Editor 27
SoloLearn – Basic Concepts – Module 1 Quiz 28
SoloLearn – Control Structures – Booleans & Comparisons 30
SoloLearn – Control Structures – if Statements 31
SoloLearn – Control Structures – else Statements 33
SoloLearn – Control Structures – Boolean Logic 35
SoloLearn – Control Structures – Operator Precedence 37
SoloLearn – Control Structures – while Loops 38
SoloLearn – Control Structures – Lists 40
SoloLearn – Control Structures – List Operations 41
SoloLearn – Control Structures – List Functions 43
SoloLearn – Control Structures – Range 46
SoloLearn – Control Structures – for Loops 47
SoloLearn – Control Structures – Simple Calculator 49

SoloLearn – Control Structures – Module 2 Quiz 50
SoloLearn – Functions & Modules – Code Reuse 52
SoloLearn – Functions and Modules – Functions 53
SoloLearn – Functions and Modules – Function Arguments 54
SoloLearn – Functions & Modules – Returning from Functions 55
SoloLearn – Functions & Modules – Comments and Docstrings 56
SoloLearn – Functions & Modules – Functions as Objects 57
SoloLearn – Functions & Modules – Modules 58
SoloLearn – The Standard Library & pip 60
SoloLearn – Functions & Modules – Module 3 Quiz 62
SoloLearn – Exceptions & Files – Exceptions 65
SoloLearn – Exceptions & Files – Exception Handling 66
SoloLearn – Exceptions & Files – finally 67
SoloLearn – Exceptions & Files – Raising Exceptions 69
SoloLearn – Exceptions & Files – Assertions 70
SoloLearn – Exceptions & Files – Opening Files 71
SoloLearn – Exceptions & Files – Reading Files 72
SoloLearn – Exceptions & Files – Writing Files 73
SoloLearn – Exceptions & Files – Working with Files 74
SoloLearn – Exceptions & Files – Module 4 Quiz 75
SoloLearn – More Types – None 77
SoloLearn – More Types – Dictionaries 78
SoloLearn – More Types – Dictionary Functions 80
SoloLearn – More Types – Tuples 81
SoloLearn – More Types – List Slices 82
SoloLearn – More Types – List Comprehensions 83
SoloLearn – More Types – String Formatting 84
SoloLearn – More Types – Useful Functions 85
SoloLearn – More Types – Text Analyzer 87
SoloLearn – More Types – Module 5 Quiz 88
Continue SoloLearn (if you want) 90
Course Coding Assignments 91
Senior Project Proposal 92

Course Introduction

Welcome to grade 10 computer studies (ICS2O). You will be learning how to program a computer using the Python programming language, version 3.6.

Watch the Course Introduction video at https://youtu.be/IrC45mwcUt8 and then read the following.

Mark Breakdown

Category	**Percent**
Term Work Python Workbook, Coding Challenges, Written Tests	70%
Senior Project Proposal Write a proposal document for a project to be completed in grades 11 and 12. It can be a game, mobile app, web application, robotics application, etc.	10%
Exam	20%

Email

The majority of communication between you and I will be via your school email account. Make sure you check your school email account at the start and at the end of each period.

GitHub

You will be using an industry-standard code storage and versioning system called GitHub in this course. All your work will be pushed (uploaded) to GitHub daily, so I have constant access to your work. When you want me to mark something, you simply must make sure your work is pushed to GitHub and that you send me a notification email.

I have created a private repository (storage area) on GitHub for you. To access it, you need to have a GitHub account. So, head over to https://github.com/ and create an account. Then email me your GitHub username so I can add you as a collaborator on the repository.

NOTE: You can access GitHub from school and from home.

Follow along with https://youtu.be/hMymT4qJB2c to setup and use GitHub.

Course Software

All the software in this course is free to download and use. This means you can set up your home computer just like your one at school.

Daily Workflow

Follow this workflow whenever you work on this course (at school or at home):

1. Log into your computer.
2. Pull your repository from GitHub.
3. Check email.
4. Work on this workbook, do coding challenges, work on final project, etc.
5. Commit and push your repository to GitHub.
6. Send me any required notification emails.
7. Log off your computer.

Mastery System

The best way to learn to program is to keep working away at it until you've mastered the basics. That's why I require you to keep redoing your assignments in this course until they are perfect. I mark everything on a 4-point scale:

Level	**Description**
1	You are just beginning to understand a concept.
2	Your understanding is developing.
3	You are proficient, but still make some mistakes.
4	You have mastered a concept and very rarely make mistakes.

All lessons in this workbook must be completed to LEVEL 4 before you can go onto the course Coding Challenges. Any given Coding Challenge must be mastered to LEVEL 4 before you can write the Coding Test associated with that Coding Challenge.

Python Workbook

You will learn the basics of the Python programming language by completing the lessons in this workbook. When you have completed a lesson in this workbook, ensure it is pushed to GitHub and then send me a notification email.

You must complete all lessons in this workbook to LEVEL 4 before going on to the course Coding Challenges.

Coding Challenges

There are several Coding Challenges in the course. This is where you really learn to solve problems and program! Each challenge is followed by a written test which tests the concepts learned during the challenge.

You must achieve LEVEL 4 on a Coding Challenge before you can attempt the test.

Coding Tests

After completing a Coding Challenge to LEVEL 4, you are eligible to write the Coding Test associated with the challenge. The test will cover the basic principles learned in the Coding Challenge.

You are <u>encouraged</u> (but not required) to write the Coding Tests over and over until you reach LEVEL 4.

Senior Project Proposal

In the final weeks of this course, you will be completing a proposal document for a senior project. It can be a game, mobile app, web application, desktop application, robotics application, or anything you can think of. This project will be worked on in grades 11 and 12. Even if you are not taking senior programming, you still must submit a proposal.

Final Exam

The final exam is similar to the Coding Tests and written on paper.

What is a programmer?

Watch https://youtu.be/_m4_5agpEgA and then answer the following questions:

1. What is a ***computer programmer***?

A computer programmer is someone who designs and writes computer programs to solve problems.

2. What is a ***computer program***?

A computer program is a sequence of steps which can be carried out by a computer to automate some task.

3. What is ***computer science***?

Computer science is the study of computers and programming.

4. Why is a programmer like a mathematician?

Programmers use formal language to convey ideas and carry out computations.

5. Why is a programmer like an engineer?

Programmers design, build, and test complex solutions to real world problems.

6. Why is a programmer like a scientist?

Programmers observe the behavior of complex systems, form hypotheses, and test predictions.

7. List and describe *three* important aspects of problem solving.

Break a complex problem down into simpler ones.
Think creatively about solutions.
Express solutions clearly and accurately.

What is Python?

Watch https://youtu.be/6eKbYdESQoQ and then answer the following questions:

1. What is an algorithm?

An algorithm is a step by step list of instructions that if followed exactly will solve the problem under consideration.

2. Write an algorithm for making a peanut butter and jelly sandwich. Don't forget any steps!

1. Get bread.
2. Get peanut butter.
3. Get jelly.
4. Get a butter knife.
5. Using the butter knife, spread the peanut butter on one side of the bread.
6. Using the butter knife, spread the peanut butter on the other side of the bread, facing the one with the peanut butter.
7. Put the two sides with the peanut butter and jelly together.

3. What is a programming language?

A programming language is a set of commands, instructions, and other syntax which can be used to issue commands to a computer.

4. Compare high-level and low-level languages. How are they similar? How are they different?

High level languages are. readable by humans and therefore make programs much easier to read and write. Low level languages are not readable to humans. Both languages are used by computers to run programs but high level can be read by humans while low level cant.

5. List *five* high-level languages and briefly describe the uses of each one.

Python, is a scripting language.
Pascal, used to teach programming languages.
c++, is used for scripting.
java, used to create applications on a programmers computer.
ruby, helps developers build websities

6. What must be done in order to run a program written in a high-level language?

Translate it into lower level language.

7. Why don't programmers generally program in machine language? Why use a high-level language?

Because it is longer and harder to read and understand.

8. What is the function of the interpreter?

The interpreter translates the python code to machine code.

Set up your development environment

Watch https://youtu.be/YJqJOgIltic and then answer the following questions:

1. Which version of Python are we using in this class?

Version 3.6

2. What is PyCharm?

Development environment used in computer programming.

3. What is the interpreter? Why does PyCharm need to know where it is on the system?

The interpreter translates the code into python. Pycharm needs to know where it is or it cant translate.

4. Why should all your code be stored on GitHub?

5. Write a program which outputs your name to the console.

More About Programs

Watch https://youtu.be/g3yrOPadK0g and then answer the following questions:

1. What is a program?

Sequence of instructions to perform a task or computation.

2. List and describe the *five* basic instructions supported by the majority of programming languages.

Debugging

Watch https://youtu.be/9Kw9NpK5LNM and then answer the following questions:

1. When programming, what is a ***bug***?

A bug is a error in the programming.

2. What is the term for the process of removing errors from software?

Debugging.

3. List the *three* types of errors and describe each one.

Syntax errors, a typing error.
Runtime errors, a error that happens when you run a program.
Logic errors, a error that makes your program not run the intended output.

SoloLearn – Basic Concepts – What is Python?

For this lesson, head over to https://www.sololearn.com/, create an account, and complete the ***What is Python*** lesson. When you have completed the lesson, answer the following questions.

1. Python is a high-level programming language. What does this mean?

It is a language that is understandable by humans.

2. List *four* popular applications of Python.

1.x, 2.x, 3.x, CPython

3. What is the interpreter?

An interpreter is a program that runs scripts written in an interpreted language such as Python.

4. How many major versions of Python are there?

3.

SoloLearn – Basic Concepts – Your First Program

Complete the ***Your First Program*** lesson at SoloLearn and then answer the following questions.

1. Write a program which will print your name to the console window. Copy and paste your code into the textbox below.

SoloLearn – Basic Concepts – Simple Operations

Complete the ***Simple Operations*** lesson at SoloLearn and then answer the following questions.

1. What happens if you try to divide by zero?

2. Predict the output of the following program. Try to figure it out before typing the program into PyCharm.

```
1    print(10 / (4 + 1) - 5)
2    print(35 - 12 / 4 - (2 * 3))
3    print(100 / (6 / 2 - 3))
```

3. What does the following error message mean?

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
```

SoloLearn – Basic Concepts – Floats

Complete the ***Floats*** lesson at SoloLearn and then answer the following questions.

1. What is a ***float***? Give *three* examples of float values.

2. What is the difference between floats and integers?

3. Predict the output of the following program. Try to figure it out before typing the program into PyCharm.

```
1  print(25 / 5 + 4.0 * 3)
2  print((7 * 6 / 2 + (100 / 20)) / 2)
3  print(-5 / 4 + -20 / 4)
```

SoloLearn – Basic Concepts – Other Numerical Operations

Complete the ***Other Numerical Operations*** lesson at SoloLearn and then answer the following questions.

1. What is the purpose of ***floor division***?

2. What is the purpose of ***modulo***?

3. Predict the output of the following program.

```
print(25 % 5 + 14.0 // 3)
print(32 // 3 - 13 % 5 ** 2)
print(1.25 ** 2 % 0.50 // 1)
```

SoloLearn – Basic Concepts – Strings

Complete the ***Strings*** lesson at SoloLearn and then answer the following questions.

1. What is a ***string***?

2. How can you include single quotes in a single quote string?

3. How can you insert a new line into a string?

4. What is an ***escape character***?

5. What is the triple quote (“””) used for?

SoloLearn – Basic Concepts – Simple Input & Output

Complete the ***Simple Input & Output*** lesson at SoloLearn and then answer the following questions.

1. What does the ***print*** function do?

2. What does the ***input*** function do?

SoloLearn – Basic Concepts – String Operations

Complete the ***String Operations*** lesson at SoloLearn and then answer the following questions.

1. What is ***concatenation***?

2. What happens if you try to add a string (str) to an integer (int)?

3. What happens when you multiply a string by an integer? What if you multiply a string by a float?

SoloLearn – Basic Concepts – Type Conversion

Complete the ***String Operations*** lesson at SoloLearn and then answer the following questions.

1. How can you convert a string to an integer?

2. How can you convert a string to a float?

3. How can you convert a number to a string?

SoloLearn – Basic Concepts – Variables

Complete the ***Variables*** lesson at SoloLearn and then answer the following questions.

1. What is a variable?

2. What are the rules of naming variables?

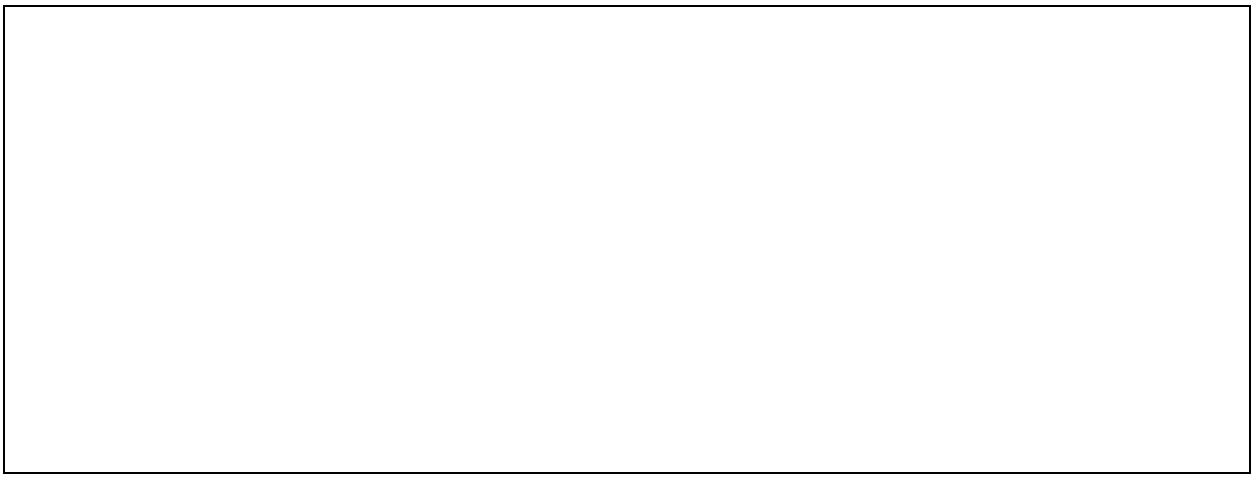
3. Variables in Python are case sensitive. What does this mean?

4. Write a program which prompts a user for their first name, then their last name, and then greets them using their full name. Here is a sample session…

```
What is your first name? Alan
What is your last name? Turing
Hello Alan Turing! Nice to meet you!
```

5. Write a program which asks the user for two numbers and then prints out the sum of the numbers. Here is a sample session…

```
Enter the first number: 5
Enter the second number: 6
5 + 6 = 11
```



SoloLearn – Basic Concepts – In-Place Operators

Complete the ***In-Place Operators*** lesson at SoloLearn and then answer the following questions.

1. What are the in-place operators used for?

2. What will the operator *= do?

3. What will the operator %= do?

4. Refactor the following code to use in-place operators.

```
a = 10
a = a + 2
a = a * 10
a = a - 5
a = a // 4
a = a / 4
a = a % 2

print(a)
```

Refactoring: https://searchmicroservices.techtarget.com/definition/refactoring

5. Write a program which asks the user for their age and then tells them how old they will be in 5 years.

```
How old are you? 15
You will be 20 in 5 years.
```

SoloLearn – Basic Concepts – Using an Editor

Complete the ***Using an Editor*** lesson at SoloLearn and then answer the following questions.

1. What are the advantages of using a code editor (e.g. PyCharm) over just using the Python console?

SoloLearn – Basic Concepts – Module 1 Quiz

Complete the ***Module 1 Quiz*** lesson at SoloLearn and then answer the following questions.

1. What is the output of the following code if the user enters the number 5?

```
x = int(input("Enter a number: "))
result = x ** x * 2 // x % 50
print(result)
```

2. What is the output of the following code if the user enters the character * and the number *8*?

```
c = input("Enter a character: ")
n = input("Enter a number: ")
n = int(n)
print(c * n)
```

3. The following code is supposed to calculate the area of a rectangle, but there are errors in it.
    a. Determine each of the errors and the type (syntax, runtime, or logic). Specify the line number and the errors details.
    b. Fix the code.

```
1 print("Hello! I will calculate the area of a rectangle!")
2 l = input("Enter the length in metres: ")
3 w = input("Enter the width in metres: ")
4 area = l / w
5 print("The area of the rectangle is  + str(Area) + " squared metres.")
```

SoloLearn – Control Structures – Booleans & Comparisons

Complete the ***Booleans & Comparisons*** lesson at SoloLearn and then answer the following questions.

1. What are the possible values of a Boolean variable?

2. State the purpose of each of these Boolean operators: ==, !=, >, <, <=, and >= .

3. What is the difference between the = operator and the == operator?

4. Can strings be compared using Boolean operators?

SoloLearn – Control Structures – if Statements

Complete the ***if Statements*** lesson at SoloLearn and then answer the following questions.

1. What is the purpose of the if statement?

2. What is ***indentation*** and why is it important in Python?

3. Write a program which prompts the user for two numbers and then tells the user which number is greater.

4. Find and correct the errors in the following program. State the type of each error found (syntax, runtime, or logic).

```
"""
This program determines whether you can see an R-rated movie.
""

age = input("How old are you? ")

if age < 18
    print("Sorry! You are not old enough to see this movie!")

if age > 18:
print("You are old enough to see this movie! Enjoy!")
```

SoloLearn – Control Structures – else Statements

Complete the ***else Statements*** lesson at SoloLearn and then answer the following questions.

1. What is the else statement used for?

2. What is the elif statement used for?

3. Write a program which prompts the user for two letters and then informs the user which letter comes first in the alphabet. Remember, strings (and characters) can be compared using the comparison operators.

4. Find and correct the errors in the following program. State the type of each error found (syntax, runtime, or logic).

```
"""
This program determines if an integer entered by the user is
positive, negative, or zero.
"""

n = int("Enter a number: ")

if n < 0:
    msg = "The number is negative."
elif n > 0
    msg = "The number is positive."
else:
msg = "The number is zero."

print(Msg)
```

SoloLearn – Control Structures – Boolean Logic

Complete the ***Boolean Logic*** lesson at SoloLearn and then answer the following questions.

1. Determine the output of the following program.

```
a = True
b = True
c = False

print(a or b)
print(a and b)
print(a or c)
print(not a)
print(not a and c)
print(not(a and c))
print(not a or not c)
```

2. Write a program which determines if an animal is a reptile. It should ask the user the following questions:

QUESTION:	CORRECT ANSWER:
How many legs does it have?	4
Does it lay eggs?	yes
Does it have scales?	yes
Is it warm-blooded?	no
Does it have lungs?	yes

If the user answers all the questions correctly, then the program should inform them that the animal is probably a reptile. If any of the answers are wrong, it should inform them that the animal is probably not a reptile.

SoloLearn – Control Structures – Operator Precedence

Complete the ***Operator Precedence*** lesson at SoloLearn and then answer the following questions.

1. What does it mean for an operator to have “higher precedence” than another operator?

2. Determine the output of the following program.

```
a = 10
b = 7
c = 3

print(a + c > b)
print(a // b * c == 3 % 4)
print((c * b) % 2 != 0 and c ** 2 // 4 <= 7 % 2)
print(not((a - b) // 10 == 0 or a ** c % 2 != 1))
```

SoloLearn – Control Structures – while Loops

Complete the ***while Loops*** lesson at SoloLearn and then answer the following questions.

1. When might you use a ***while loop*** in your code?

2. What is ***iteration***?

3. What does the ***break*** statement do?

4. What does the ***continue*** statement do?

5. Write a program which calculates the sum of the numbers from 1 to a number entered by the user. So if the user enters 5, the program will output 15. If the user enters 6, the program will output 21. And so on.

SoloLearn – Control Structures – Lists

Complete the ***Lists*** lesson at SoloLearn and then answer the following questions.

1. What are lists used for?

2. What types of values can a list contain?

3. What is an ***IndexError*** and how is it caused?

4. Write a program which prompts the user for 10 floats, stores them in a list, and then calculates the average of the numbers entered.

SoloLearn – Control Structures – List Operations

Complete the lesson ***List Operations*** at SoloLearn and then answer the following questions.

1. How are strings like lists?

2. How can two lists be added together to create one list?

3. Determine the output of the following program.

```
nums = [0, 1, 2, 3, 4, 5]

i = 5
while i > 0:
    nums[i] += i % 2
    i -= 1

bools = []
while i < 5:
    bools = bools + [i in nums]
    i += 1

print(bools)
```

4. Write a program which determines whether a color entered by the user is one of the colors of the visible light spectrum (*red*, *orange*, *yellow*, *green*, *blue*, *indigo*, or *violet*). The program should repeat until the user enters the word *exit*.

SoloLearn – Control Structures – List Functions

Complete the lesson ***List Functions*** at SoloLearn and the answer the following questions.

You can find a more complete discussion of *list* functions and methods at:
https://www.tutorialspoint.com/python3/python_lists.htm

1. A method is a function which belongs to an object. For example, the *append method* belongs to the *list* object. What does the append method do?

2. A function is a useful piece of code (like a mini-program) which performs a specific task. What is the *len* function used for?

3. What is the purpose of the *insert* method of the *list* object?

4. What does the *index* method of the *list* object do?

5. State the uses of the following *list* functions and methods: *max, min, count, remove,* and *reverse.*

6. Write a program which accepts a list of numbers from the user and then determines the minimum and maximum values in the list. Use the *min* and *max* functions.

7. Rewrite your program from question #6, but this time you cannot use the *min* and *max* functions.

SoloLearn – Control Structures – Range

Complete the lesson ***Range*** at SoloLearn and then answer the following questions.

1. What does the *range* function do?

2. What does the *list* function do?

3. Use the range function to create a list of odd numbers from 0 to 100.

SoloLearn – Control Structures – for Loops

Complete the ***for Loops*** lesson at SoloLearn and then answer the following questions.

1. When is the *for loop* used?

2. Write a program which accepts a message from the user and a number of times to repeat the message. It should then print the message to the console the specified number of times.

3. Write a program to calculate the Fibonacci sequence. The user should enter the number of terms to be calculated.

Reference links:
https://www.mathsisfun.com/numbers/fibonacci-sequence.html

SoloLearn – Control Structures – Simple Calculator

Complete the ***Simple Calculator*** lesson at SoloLearn and then answer the following question.

1. Complete the Simple Calculator program.

SoloLearn – Control Structures – Module 2 Quiz

Complete the ***Module 2 Quiz*** lesson at SoloLearn and then answer the following questions.

1. What is the output of the following code?

```
for i in range(2, 10, 2):
    j = 0
    while j < 4:
        print(i * j)
        j = j + 1
```

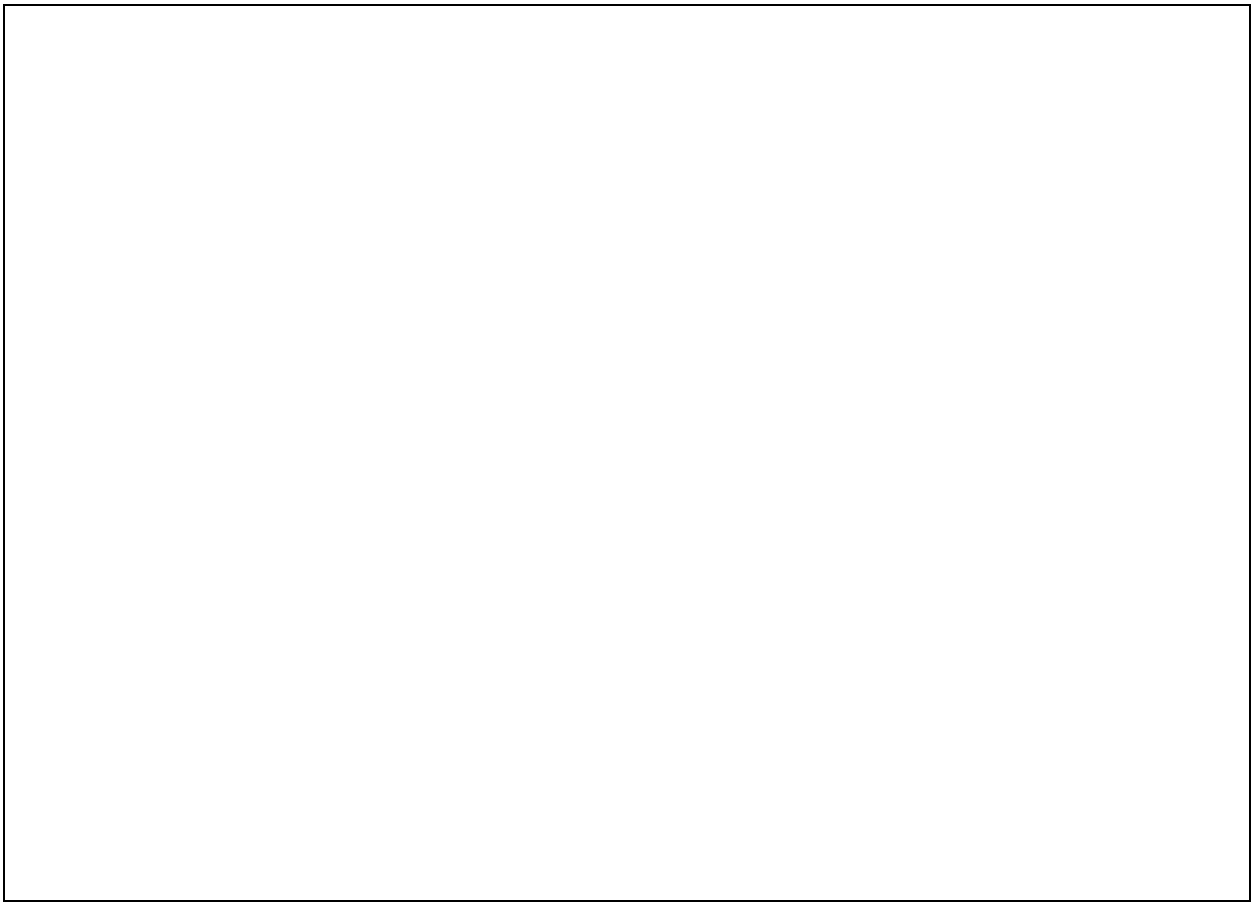
2. Write a program which prints all divisors of a positive integer entered by the user.

3. Debug the following program.

```
"""
This program calculates the volume of a number of boxes in
cubic centimetres.
"""

# Box data is stored in a list of lists.  Each sublist contains
# the length, width, and height of the box.
box_data = [
    [8.5, 10.0, 3.5]
    [7.0, 8.0, 4.0]
    [12.0, 12.0, 12.0]
    [9.5, 11.5, 2.0]
    [9.0, 6.0, 1.5]
]

for box in box_data
    length = box[1]
    width = box[2]
    height = box[3]
    volume = Length + Width + Height
    print(volume + " square metres")
```



SoloLearn – Functions & Modules – Code Reuse

Complete the lesson ***Code Reuse*** lesson at SoloLearn and then answer the following questions.

1. What is the ***DRY Principle***? Why is it important?

2. How do functions support the DRY Principle?

3. List and state the use of *five* functions you have already encountered in Python.

SoloLearn – Functions and Modules – Functions

Complete the ***Functions*** lesson at SoloLearn and then answer the following questions.

1. What does the ***def*** keyword do?

2. Write a function which prints the message “Hello, world!” to the console 10 times.

SoloLearn – Functions and Modules – Function Arguments

Complete the ***Function Arguments*** lesson at SoloLearn and then answer the following questions.

1. What is a ***function argument***?

2. Write a function called ***my_print*** which accepts a message as an argument and then prints the message to the console window.

3. Can variables defined in functions be accessed from outside the function? Give an example with your answer.

4. What is a ***NameError***?

SoloLearn – Functions & Modules – Returning from Functions

Complete the ***Returning From Functions*** lesson at SoloLearn and then answer the following questions.

1. What is the purpose of the ***return*** statement?

2. Write a function which accepts two integers and returns the greater integer.

3. Write a function which accepts a list of floats and returns the average of the list.

SoloLearn – Functions & Modules – Comments and Docstrings

Complete the ***Comments and Docstrings*** lesson at SoloLearn and then answer the following questions.

1. Why is it important to comment your code?

2. What is a docstring?

3. How can you write a multiline string in your code?

SoloLearn – Functions & Modules – Functions as Objects

Complete the ***Functions as Objects*** lesson at SoloLearn and then answer the following questions.

1. Create a function that accepts two strings. If the second string is contained in the first, the function returns True. If the second string is not contained in the first, the function returns False. Then, assign it to a variable and call it using the variable.

SoloLearn – Functions & Modules – Modules

Complete the ***Modules*** lesson at SoloLearn and then answer the following questions.

1. What is a ***module***?

2. Which module helps you to generate random numbers?

3. Which module contains helpful math functions and constants?

4. What causes an ImportError?

5. Explain the difference between the following three methods of importing a module.

```
1  import math
2  from math import sqrt
3  from math import sqrt as square_root
```

SoloLearn – The Standard Library & pip

Complete ***The Standard Library & pip*** lesson and then answer the following questions.

1. Describe the *three* main types of modules in Python.

2. What is the ***standard library***?

3. List and describe *five* modules contained in the standard library.

4. What is the ***Python Package Index (PyPi)***?

5. What is ***pip***?

SoloLearn – Functions & Modules – Module 3 Quiz

Complete the ***Module 3 Quiz*** lesson at SoloLearn and then answer the following questions.

1. Define a function which accepts a list of floats are returns the *mode* of the list.

Reference links:
https://www.mathsisfun.com/definitions/mode.html

2. Define a function which accepts a list of integers and returns a list of the integers squared.

3. Write a program which simulates the rolling of a 6-sided die. It should ask the user how many rolls they wish to do, then it should output the requested number of die rolls.

4. Predict the output of the following program.

```
def f(x):
    if x < 6:
        return 0
    elif 6 < x < 10:
        return 1
    else:
        return x


x = 15
for i in range(x):
    print(f(i))
```

5. Debug the following program.

```
"""
This program determines the area of triangles.  The bases and heights
of the triangles are stored in a list.  Each pair of numbers are the
base and height of a triangle.
"""


def area():
    return 0.5 * b * h


data = [1.0, 1.0, 2.2, 3.3, 4.0, 7.4, 4.2, 3.2, 8.0, 6.1]

for i in range(0, len(data), 1):
    print(area(data[0], data[1]))
```

SoloLearn – Exceptions & Files – Exceptions

Complete the ***Exceptions*** lesson at SoloLearn and then answer the following questions.

1. What is an ***exception***?

2. List and describe *six* common exceptions.

SoloLearn – Exceptions & Files – Exception Handling

Complete the ***Exception Handling*** lesson at SoloLearn and then answer the following questions.

1. What is the purpose of the ***try/except*** statement?

2. As a programmer, why might you want to handle your exceptions?

3. Predict the output of the following program.

```
x = 10
while x > 0:
    try:
        print(x / (x - 1))
    except ZeroDivisionError:
        print("Division by zero!")
    x -= 1
```

SoloLearn – Exceptions & Files – finally

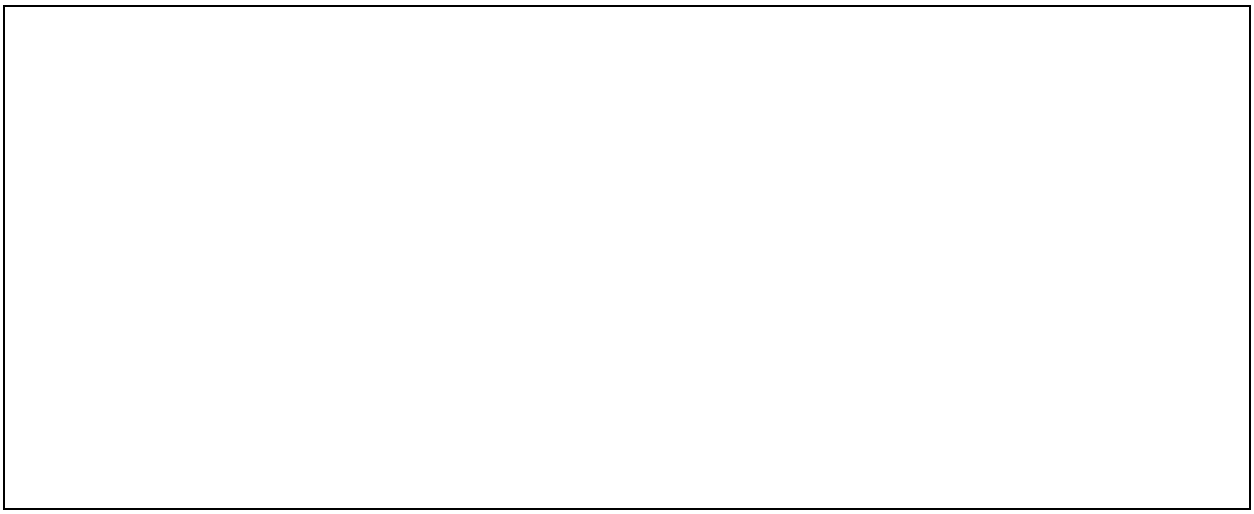
Complete the ***finally*** lesson at SoloLearn and then answer the following questions.

1. What is the purpose of the ***finally*** statement?

2. Write a program which calculates the square root of a number entered by the user. If they enter a negative number, the program should inform them that they must enter zero or a positive number. The program should say "Goodbye!" no matter what.

3. Determine the output of the following program.

```
nums = [1, -1, 6, -6, -1, -3, 0, 0, 9, 5, -5]
n = len(nums) - 2
while n > 0:
    try:
        print(10 // (nums[n + 1] + nums[n]))
    except:
        print("Invalid")
    finally:
        n = n - 1
```



SoloLearn – Exceptions & Files – Raising Exceptions

Complete the ***Raising Exceptions*** lesson from SoloLearn and then answer the following questions.

1. What is the purpose of the ***raise*** keyword?

2. Why would a programmer want to raise their own exceptions?

3. Rewrite your square root calculator from the last lesson so that it raises a custom exception when the user enters a negative number and BEFORE it attempts to calculate the square root.

SoloLearn – Exceptions & Files – Assertions

Complete the ***Assertions*** lesson at SoloLearn and then answer the following questions.

1. What is an ***assertion***?

2. When do programmers often use assertions?

3. What will happen to the program if an assertion is not handled?

4. Write a program that asks for the user's name. Use an assertion to make sure the user doesn't enter an empty string. If they enter an empty string, use an assertion to end the program.

SoloLearn – Exceptions & Files – Opening Files

Complete the ***Opening Files*** lesson at SoloLearn and then answer the following questions.

1. Which function is used to load data from files?

2. What is the first argument passed to the ***open*** function?

3. What do the letters r, w, a, and b mean when passed as the second argument to the open function?

4. How do you close a file after it has been opened?

SoloLearn – Exceptions & Files – Reading Files

Complete the ***Reading Files*** lesson at SoloLearn and then answer the following questions.

1. What does the ***read*** method of the file object do?

2. What does the ***readlines*** method of the file object do?

3. Write a program which will load numbers from a file and determine: the average, the maximum value, and the minimum value. Write the output to the console window.

Input file: https://goo.gl/8sMd1f

SoloLearn – Exceptions & Files – Writing Files

Complete the ***Writing Files*** lesson at SoloLearn and then answer the following questions.

1. Which method is used to write data to a file?

2. What happens to the content of a file if it is opened in write mode?

3. Write a program which generates 1000 random numbers between 1 and 100 and writes them to a file.

SoloLearn – Exceptions & Files – Working with Files

Complete the ***Working with Files*** lesson at SoloLearn and then answer the following questions.

1. Why is a good idea to use a try/except/finally block when opening files?

2. What does the ***with*** statement do?

3. Write a program which imports a list of names and outputs a file containing only the names starting with the letter C.

Input file: https://goo.gl/NSp17A

SoloLearn – Exceptions & Files – Module 4 Quiz

Complete the ***Module 4 Quiz*** lesson at SoloLearn and then answer the following questions.

1. Predict the output of the following program.

```
for n in range(10, -10, -2):
    try:
        assert n > -3
        print(n)
    except AssertionError:
        continue
```

2. Write a program which calculates the volume of rectangular prisms. The input file contains the length, width, and height of many rectangular prisms. Each line contains the data for a prism. You will find the ***split*** string method useful. Your program should output a file with the volumes of each prism.

Input file: https://goo.gl/o1aTL7
split method reference: https://www.pythonforbeginners.com/dictionary/python-split

SoloLearn – More Types – None

Complete the ***None*** lesson at SoloLearn and then answer the following questions.

1. What is the ***None*** keyword used for?

2. What is the value of None when converted to a Boolean?

3. Predict the output of the following program.

```
print(bool(None))
print(bool(0))
print(bool(0.0))
print(bool([]))
```

SoloLearn – More Types – Dictionaries

Complete the ***Dictionaries*** lesson at SoloLearn and then answer the following questions. Read more about Python dictionaries at https://www.w3schools.com/python/python_dictionaries.asp

1. What is a ***dictionary***?

2. Predict the output of the following program.

```
fruits = {
    "apples": 10,
    "oranges": 5,
    "lemons": 5,
    "strawberries": 10}

fruits["apples"] += 1
fruits["oranges"] -= 2
fruits["lemons"] *= 2
fruits["strawberries"] //= 2

print(fruits["apples"])
print(fruits["oranges"])
print(fruits["lemons"])
print(fruits["strawberries"])
print(fruits)
```

3. What is an ***immutable object***? Why can only immutable objects be used as keys in a dictionary?

SoloLearn – More Types – Dictionary Functions

Complete the ***Dictionary Functions*** lesson at SoloLearn and then answer the following questions.

1. Write a program which counts the instances of each animal in an input file. Use a dictionary to keep track of the data. Your output should list each animal and the number of times it appears in the list.

Input file: https://goo.gl/X3e6td

SoloLearn – More Types – Tuples

Complete the ***Tuples*** lesson at SoloLearn and then answer the following questions.

1. What is a ***tuple***? How are they similar to lists? How are they different?

2. What happens if you attempt to change a value in a tuple?

3. Why would a programmer choose to use a tuple instead of a list?

4. Define a function which calculates the distance between two points on a graph. The input to the function should be the two coordinate pairs in tuple form. The output should be the distance between the points. If the input is incorrect, the function should raise an exception.

The Distance Formula: http://www.mathwarehouse.com/algebra/distance_formula/index.php

SoloLearn – More Types – List Slices

Complete the ***List Slices*** lesson at SoloLearn and then answer the following questions.

1. What is a ***list slice***?

2. Predict the output of the following program.

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

print(letters[1:3])
print(letters[3:1])
print(letters[3:-3])
print(letters[2:])
print(letters[:5])
print(letters[:])
print(letters[1:5:2])
print(letters[-6:-4])
```

3. Define a function which determines the number of times a given substring appears in a string. The function should accept two arguments: the string to be searched and the character to be searched for. The function should return the number of times the character appears in the string. HINT: slicing can be used on strings too!

SoloLearn – More Types – List Comprehensions

Complete the ***List Comprehensions*** lesson at SoloLearn and then answer the following questions.

1. What is a ***list comprehension***?

2. Write a program which produces a list of random integers. Use a list comprehension to create a list of 100 random integers, each between 0 and 100.

SoloLearn – More Types – String Formatting

Complete the ***String Formatting*** lesson at SoloLearn and then answer the following questions.

1. Write a program which prompts the user for their name and then uses string formatting to output their name in a fancy asterisk frame. Like this…

```
***************
* Alan Turing *
***************
```

SoloLearn – More Types – Useful Functions

Complete the ***Useful Functions*** lesson at SoloLearn and then answer the following questions.

1. State the purpose of each of the following string methods: ***join, replace, startswith, endswith, upper, lower,*** and ***split***.

2. State the purpose of the following numeric functions: ***min, max, abs,*** and ***sum***.

3. State the purpose of the following list functions: ***all, any,*** and ***enumerate***.

SoloLearn – More Types – Text Analyzer

Complete the ***Text Analyzer*** lesson at SoloLearn and then answer the following questions.

1. Write a program which loads a file containing some text and then outputs a file containing the count of each letter appearing in the text. Try to make your output match mine exactly!

Input file: https://goo.gl/mMRhJx
Sample output file: https://goo.gl/hn3Ynu

SoloLearn – More Types – Module 5 Quiz

Complete the ***Module 5 Quiz*** lesson and then answer the following questions.

1. When would a programmer use a tuple instead of a list?

2. How would you use slicing to reverse a list of numbers?

3. What does the term *immutable* mean? Are lists immutable? What about dictionaries? What about tuples?

4. What is the return value of a function with no return statement?

5. Predict the output of the following program.

```
x = ([1, 2], [3, 7], [7, 8], [9, 3])

print(x[0][0:2])

x[1][1] += 2
print(max(x[1][0], x[1][1]))

x[2][0:2] = [x[2][1] + 1, x[2][0] + 2]
print(x[2])

x[-2][0] += max(x[2][0], x[3][1])
print(x[2])
```

Continue SoloLearn (if you want)

I do not require you to complete anymore of the SoloLearn course. However, you will learn a lot by completing it on your own time. And you get a fancy certificate too!

Now it's time to go onto the course coding assignments. You will be implementing what you learned in the SoloLearn course while working on these assignments.

Course Coding Assignments

The coding assignments are available in the course Google Drive. Remember, each assignment has a follow-up test, and you can only write the test after you have achieved **LEVEL 4** on the assignment.

Coding Assignments: https://goo.gl/Ag7dyY

Senior Project Proposal

Now that you have gained some serious programming skill, it's time to think about what you really want to make. Is it a game? A mobile app? A website? A robotics application? The Senior Project is your chance to work on your first big project as a developer.

This year, you will complete the first draft of the software design document (SDD). The SDD is a description of how you are going to solve a problem. It outlines the features of a piece of software and serves as a guide while you develop the software. Basically, it the most useful tool for making sure the right work gets done on a project.

Check the course Google Drive for the SDD templates. If you don't find a SDD that suits your needs, then send me an email and I can make up a custom one for you. There are SDDs for each of the main project types.

SDD Templates: https://goo.gl/vuks7E