|  |
| --- |
| Logbook for ISD |
| Your name and student number |
|  |
|  |
|  |

Logbook for ISD

Your name and student number

Contents

[Introduction 2](#_Toc431296390)

[Week 1 3](#_Toc431296391)

[Exercises 1 3](#_Toc431296392)

[Exercises 2 3](#_Toc431296393)

[Exercises … 3](#_Toc431296394)

[Week 2 4](#_Toc431296395)

[Exercises 1 4](#_Toc431296396)

[Exercises 2 4](#_Toc431296397)

[Exercises … 4](#_Toc431296398)

[Week 3 5](#_Toc431296399)

[Exercises 1 5](#_Toc431296400)

[Exercises 2 5](#_Toc431296401)

[Exercises … 5](#_Toc431296402)

[Week…13 6](#_Toc431296403)

[Exercises 1 6](#_Toc431296404)

[Exercises 2 6](#_Toc431296405)

[Exercises … 6](#_Toc431296406)

# Introduction

A brief introduction to what you have done within the module and how your experience was with the exercises and the overall module. Probably up to half a page.

# **Week 1**

## Question: 1 - 1. What is a code repository (often also called version control system) used for?

A code repository is a place to store a large amount of information like a file archive and web hosting facility such as for example: source codes (basically programs) or documents. A code repository will allow software or web pages to hold data(information) which can either be access publicly or privately. Which allows multiple users to work on a same file, code, document.

## Question: 2 - 2. Why is it advantageous to use a code repository?

A code repository is rather advantageous to use due to the way it allows a user to store information that may be duplicated and then given access to other users to use or modify otherwise for the simple reason that it allows to store private information that can be modified at any given time. It can hold a significant amount of information on a single server making the access to the stored information easy and convenient.

## Question: 3 - 3. Describe the different “layers” of Software that exist on a typical computer and explain why there are different layers of software.

A computer has three main “layers” of software:

The first layer would be the hardware (the hardware isn’t a software but it is fundamental for a software to operate properly. It is the physical part of the computer which allows the operating system to run, it also represent the physical components of the computer such as the memory, RAM…).

The second is the system software (a system software is some sort of computer program that is created to run the hardware of a computer it then allows other software to run on the computer).

The final third layer would be the application software (the application software are programs that we use on a day to day basis, it is programs that the user directly interacts with and chooses if it wants to use it or not. For example, it can be word processing or computer games or spreadsheets).

Those three layers of software are all linked together in order for each and every one of them to run as expected they are all connected to the hardware which allows a computer to work.

## Question: 4 - 4. Describe what an algorithm is and explain why it is a useful “tool” to translate from a human level problem (we can think of) to a computer program.

An algorithm is way to process a problem, which can also be seen as a set of rules that leads to calculations or other means of problem solving operations. For example, we may refer a cooking recipe as an algorithm, because in order to make the dish we have a to write down in a specific order and explain how to do it in a specific way and then it must be done exactly has written. Algorithms are mostly used by a computer because a computer cannot do things by itself. A computer requires a process and a set of rules in order to make to do something or some sort of calculation. That is why an algorithm is an ideal tool to translate from a human level problem to a computer program. Since we can code down in a specific and simple way so that the computer can run our way of thinking.

***Week 2***

Some overview of the topics covered by the lecture and the exercises. Not too much, may be a paragraph.

Question 1: Write an algorithm that describes how to make scrambled eggs, try to use control words, like IF, WHEN, UNTIL, WHILE, WAIT, AND, OR.

**Algorithm:** *Scrambled eggs*

Step 1: Get eggs, Milk, Salt, Pepper.

Step 2: Open eggs in a bowl AND beat the eggs UNTIL the egg white AND yellow mix together AND make one substance.

Step 3: Add milk AND salt AND pepper to bowl #(containing the whisked eggs)

Step 4: Turn on stove to medium OR higher heat

Step 5: Put a frying pan on stove AND add butter OR cooking spray

Step 6: Pour in egg mixtures AND immediately reduce heat to medium OR low

Step 7: Cook egg mixtures UNTIL eggs are thickened AND no visible liquid egg remains

Step 8: Wait for egg mixture to be thickened IF thickened THEN remove thickened egg from frying pan, IF NOT then eggs are too dry then throw dry eggs in bin

Question 2: Is Idle (the Python language shell) an Interpreter or an Compiler or both? Explain your answer.

What is **IDLE** what does it stand for: **IDLE** stands for *Integrated DeveLopment Environment* and it is the standard Python development environment. Now we may ask ourselves what is an "Interpreter" and an "Compiler".

**An Interpreter** is a program that reads and executes codes. Common interpreters include software languages *Perl, Python and Ruby* interpreters, which execute their codes respectively.

**An Compiler** is a program that converts high-level instructions (human problems for example) into a machine-code or low-level so that they may be read and executed by a computer.

So the Python language shell also known as IDLE is more of an interpreter then a compiler because you can type directly Python code into the shell and whenever you complete a code fragment you can execute it. Python is known for being a high level language so even if some code fragments looks like our language the Python language shell still requires specific codes that some users may not understand. So it is not a compiler but does consist of a just-in-time compiler.

Question 3: Write a command in the Idle shell that says “Hello world”

The command in the idle shell that would print "Hello world" (Program "Hello World1") is:

* Print("Hello world");

By pressing the "run" will cause the following to be displayed

* Hello world

Question 4: Write a program that produces the following output:

Hello World

I am in my ISD class right now

Program "Hello World 2":

print ("Hello World I am in my ISD class right now");

Or:

print ("Hello World");

print(" I am in my ISD class right now");

Question 5: Write a program that asks the user for his/her name and produces an output like: Hi there, what is your name?

>User input to be read<

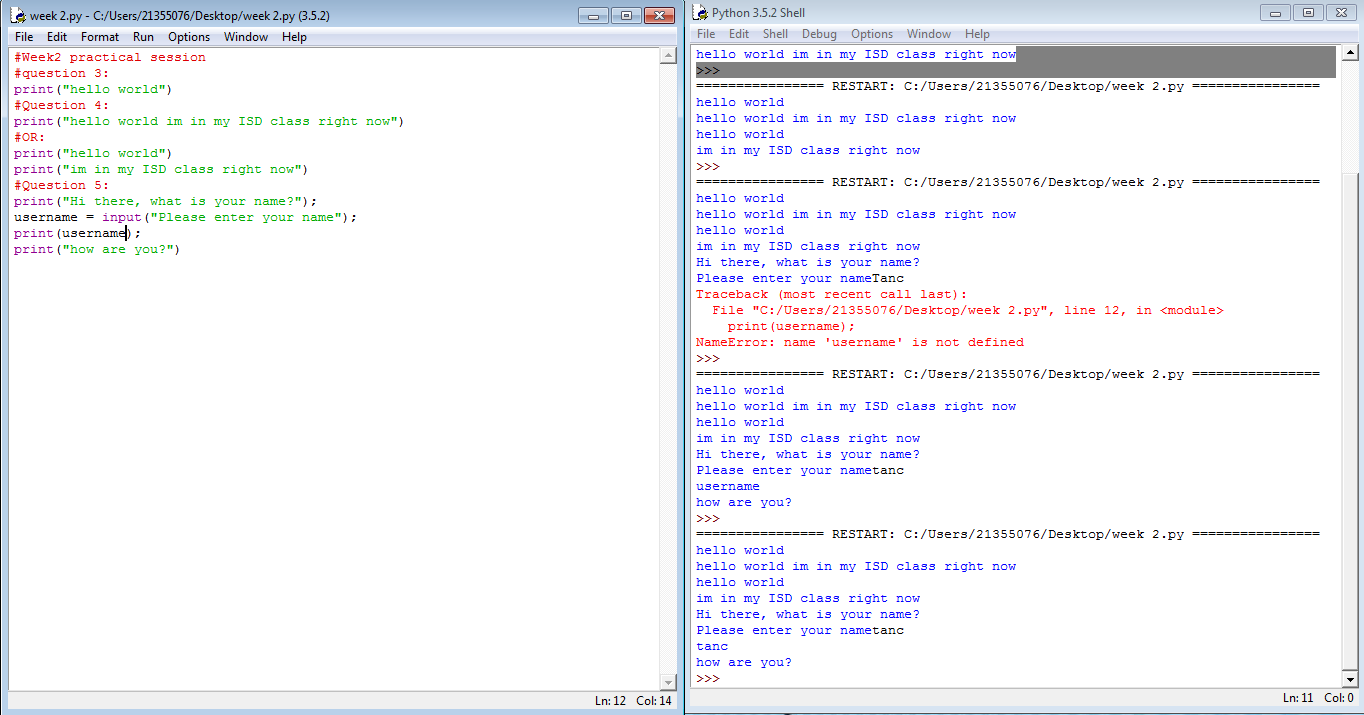
Hello

“User name”

How are you?

print("Hi there, what is your name? ");

username = input("Please enter your name");

print(username”) print("Howareyou?username");

**Week 3**

Some overview of the topics covered by the lecture and the exercises. Not too much, may be a paragraph.

## Question 1: Write a program that asks for two numbers (Python has all the basic mathematical functions in place, like +,- etc.), adds them up and displays the result.

mynumber1 = int(input("enter a number1"));

mynumber2 = int(input("enter a number2"));

result = mynumber1 + mynumber2;

print("num1 plus num2=",result);

Question 2: Answer the questions by implementing the code and run it.

a) What will the output be from the following code?

num = 4

num\*=2

num1=num+2

num1+=3

print(num1)

The output of the previous code is the number 13.

4x2=8

8+2=10

10+3=13

b) What do the following lines of code output? Why do they give a different answer?

0.6666666666666666

>>>

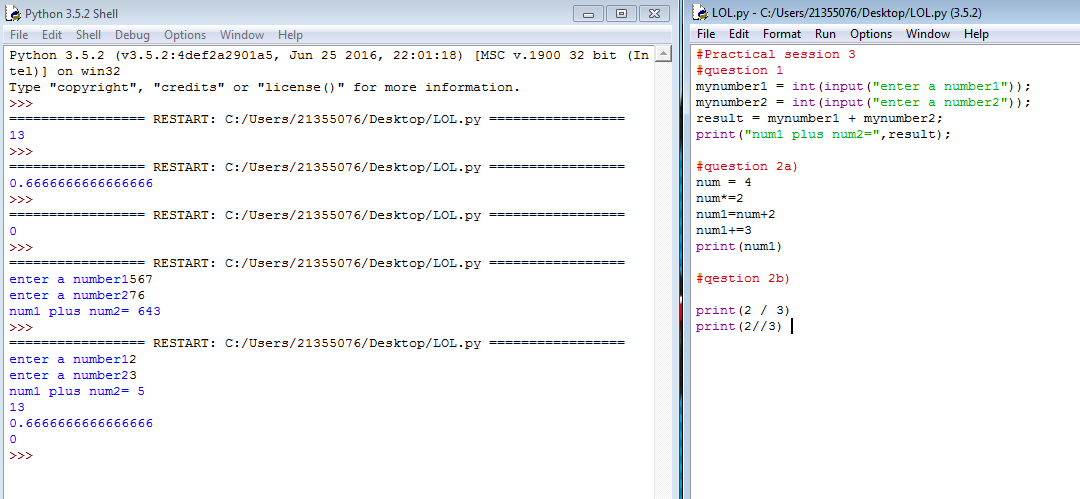
print(2 // 3)

0

>>>

The difference between “/” and “//” is the following:

“/” allows you to perform division between numbers giving you the total value after the operation is done

“//” allows you to find the quotient between numbers and will not display the reminder. 

All of the variable names below can be used. But which of these is the better variable name to use?

A a Area AREA area areaOfRectangle AreaOfRectangle

areaOfRectangle

4) Which of these variables names are not allowed in Python? (More than one might be wrong.)

apple APPLE Apple2 1Apple account number account\_number account.number accountNumber fred Fred return return\_value 5Return GreatBigVariable greatBigVariable great\_big\_variable great.big.variable

Variables names that are not allowed in Python:

1Apple, account.number, Fred, APPLE, Apple2, 5return, GreatBigVariable, great.big.variable, account number

Variables names that are allowed in Python:

apple, accountNumber, fred, return, return\_value, great\_big\_variable,

***Week 4:***

Question 1: Explain the mistake in the following code

radius = input("Radius:")

x = 3.14

pi = x

area = pi \* radius \*\* 2

*Fixed Version:*

radius = int(input("Enter the Radius of your circle: "))

x = 3.14

pi = x

area = pi \* (radius \*\* 2)

print ("The area of a circle is:\t%.2f" %area)

The mistake in the previous code was in line 1 it didn’t specify we wanted an integer, then in line 3 parenthesis were missing making the operation false and finally in line 4 there was no line 4 so no output.

Question 2: Explain the mistake in the following code:

x = 4

y = 5

a = 3(x + y)

*Fixed Version:*

x = 4

y = 5

a = 3\*(x + y)

print("The answer to a is" , a)

The mistakes were in line 3 and 4. In line 3 parenthesis were missing making the multiplication impossible to execute. In line 4 the output print wasn’t written.

Question 3: Explain the mistake in the following code:

radius = input(float("Enter the radius:"))

*Fixed Version:*

radius = float(input(“Enter the radius:”))

The mistake in this code was that float and input were in the wrong positions in python the data types have to be first in order for the code to be executable.

Question 4: Why does this code not calculate the average?

print(3 + 4 + 5 / 3)

*Fixed version:*

print((3+4+5)/3)

The code didn’t calculate the average simply because parenthesis was missing before the division.

Question 5: Consider the following code:

x = 19.93

y = 20.00

z = y – x

print(z)

The output is 0.0700000000028 Why is that so?

Improve the code so that the output is to two decimal places.

*Fixed version:*

x = 19.93

y = 20.00

z = y - x

print("the value of z is %.2f" %z)

Question 6: Find at least three compile-time errors:

int x = 2

Print (x, squared is, x \* x)

xcubed = x \*\*\* 3

*Fixed version:*

x = 2 #no need for “int”

print("x squared is" ,(x \* x)) #added quotation marks and parenthesis

xcubed = x \*\* 3 #removed one asterisk

Question 7: 7. Find two run-time errors:

from math import sqrt

X = 2

Y = 4

print(“The product of “, x, “and”, y, “is”, x + y)

print(“The root of their difference is “, sqrt(x – y))

import math

X = 2

Y = 4

print("The product of ", X, "and", Y, "is", X + Y)

print("The root of their difference is ", math.sqrt(Y - X))

Question 8: Write statements to prompt user for their name and age

Write a print statement to output:

Hello \_\_\_\_, next year you will be \_\_\_\_ years old!

my\_name = input("Enter your name")

your\_age = input("Enter your age next year")

print("Hello " + my\_name + " ,Next year you will be " + your\_age + " years old!")

Question 9: Given that radius is 2 and area is calculated as 12.5678, use string format operators to print the values of the variables radius and area so that the output looks like this:

Radius is: 2

Area is: 12.57

radius = int(input("Enter the Radius of your circle: "))

x = 3.14

pi = x

area = pi \* (radius \*\* 2)

print ("The area of a circle is:\t%.2f" %area)

#enter 2 as radius

Question 10: What are the values of the following expressions, assuming that p is 17 and q is 18?

i. p // 10 + p % 10

ii. p % 2 + q % 2

iii. (p + q) // 2

iv. (p + q) / 2.0

Assuming that p=17 and q=18 the answers are the following

1. = 8
2. =1
3. =17
4. =17.5

**Week 5:**

**Question1:** Which of the following conditions are true, if a = 13 and b = 14 ?

a) a + 1 <= b

b) a + 1 >= b

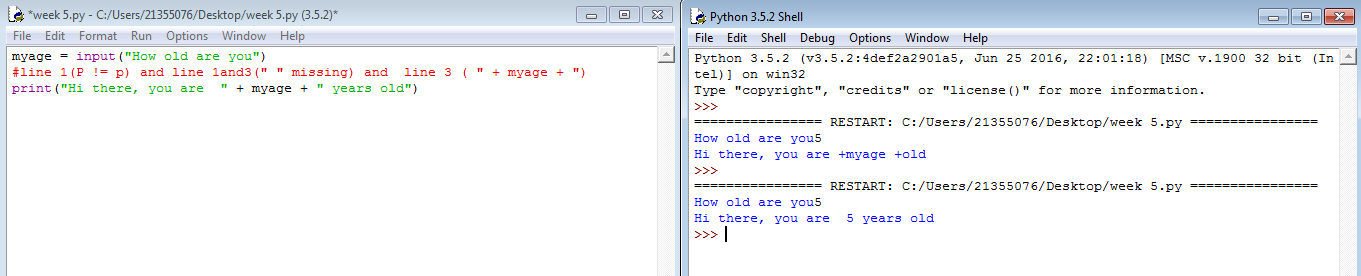
c) a + 1 != b

1. 14 <= 14 >>> True
2. 14 >= 14 >>> True
3. 14 != 14 >>> False

**Question 2:** Explain the mistake(s) in the following code:

myage = input(How old are you)

Print(Hi there, you are +myage +old)

**Question 3:** Explain why the following code won’t really add the two “numbers”:

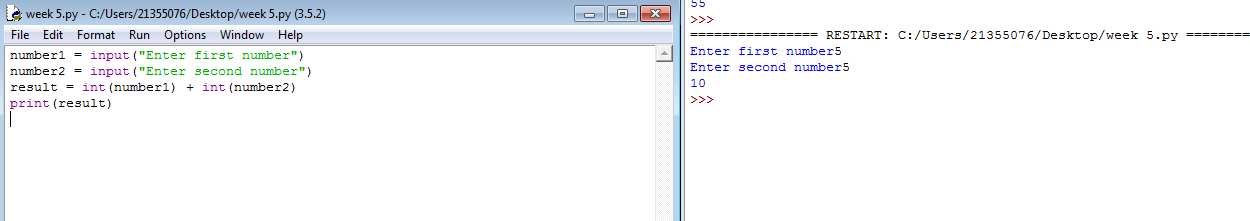
number1 = input("Enter first number")

number2 = input("Enter second number")

result = number1 + number2

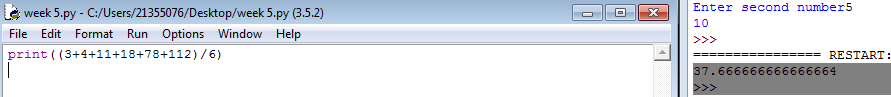
print(“The result is ”+result)

Solution:



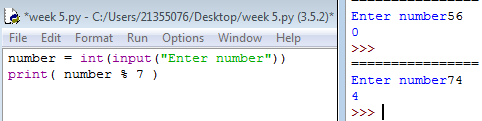
**Question 4:** Write code to calculate the average of: {3, 11, 78, 112, 4, 18} in one single line of code.

Input/Output:



**Question 5:** Write a program that asks the user for an integer number and then prints out the remainder after the number is divided by 7.

Input/Output:

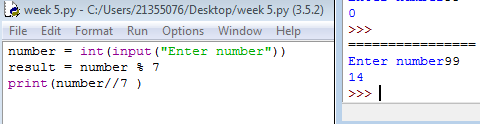


**Question 6:** Expand the above program (5.) by also printing out how often the number 7 “fits” into the number the user entered.

number = int(input("Enter number"))

result = number % 7

print(number//7 )



# Week…13

Continue this structure for the remaining weeks up until week 13

Some overview of the topics covered by the lecture and the exercises. Not too much, may be a paragraph.

## Exercises 1

Provide the exercises description and your answers. Where applicable use source code excerpts, explanations of these, represent your results, for example by showing screenshots of your program and, where applicable, display the use of your code repository (github) either by screenshots or by providing log data from your code repository.

## Exercises 2

Provide the exercises description and your answers. Where applicable use source code excerpts, explanations of these, represent your results, for example by showing screenshots of your program and, where applicable, display the use of your code repository (github) either by screenshots or by providing log data from your code repository.

## Exercises …

Example description of an exercise:

