Lesson Solutions Autumn Term year 10

GCSE (9-1) Computer Science

Pearson Edexcel Level 1/Level 2 GCSE (9-1) in Computer Science (1CP1)

# LESSON SOLUTIONS FOR AUTUMN TERM YEAR 10

# Week 1

## Lesson 1 solutions

### Activity 1.1.1

>>>print("Hello World")

Anything within the quotation marks “ ” is called a string. It is treated as a list of characters. Either single or double quotation marks can be used.

### Activity 1.1.2

If your name is Tom Jones, your program would be:

>>>print("Tom Jones")

### Activity 1.1.3

If your name is Tom Jones, your program would be:

>>>print(“Tom Jones” \* 6)

# Week 1

## Lesson 2 solutions

### Activity 1.2.1

|  |  |
| --- | --- |
| Python was released in 2010 | False – Python was released in 1989 |
| Python was named after the TV series “Monty Python’s Flying Circus” | True |
| Python is proprietary software which is expensive to buy | False – Python is open source software and freely available to everyone |
| Python was written by Mark Zuckerberg | False – Python was originally written by Van Rossum |
| YouTube is written in Python | True – Google uses Python to program many of its products |

### Activity 1.2.2

|  |  |
| --- | --- |
| **Keyboard shortcut** | **What does the keyboard shortcut do?** |
| Control c | Copy selected text |
| Control v | Paste |
| Control a | Select all |
| Control x | Cut the selected text |
| Control f | Find |
| Control n | Opens a new window |
| Control p | Prints |
| Control s | Saves |
| Control z | Undo |

### Activity 1.2.3

print("This is the end")

print("Hold your breath and count to ten”)

print("Feel the earth move and then”)

print("Hear my heart burst again”)

As above, with the lyrics changed to those from your favourite song.

### Activity 1.2.4

There is one white space in the place of the comma.

### Activity 1.2.5

|  |  |
| --- | --- |
| **Escape sequence** | **Effect** |
| \t | Tab |
| \n | New line |
| \\ | Displays \ |
| \’ | Displays ‘ |
| \” | Displays “ |

*Hint: These use back slashes \. The forward slash / has other uses in Python so make sure you use the right one.*

### Activity 1.2.6

print("Help, I need somebody\n Help, not just anybody\n Help, you know, I need someone")

# Week 2

## Lesson 1 solutions

### Activity 2.1.1

**A** = program code

**B** = pseudocode

**C** = written description

**D** = flowchart

# Week 2

## Lesson 2 activities

### Activity 2.2.1

|  |  |
| --- | --- |
| **Mathematical operator symbol** | **Operation** |
| / | divide |
| + | add |
| \* | multiply |
| \*\* | exponential |
| - | subtract |
| // | integer division |
| % | modulus (remainder after the division) |

### Activity 2.2.3

print("8 cats have 4 legs each")

print(“The cats have”,8 \* 4,”legs in total”)

print(“A farmer with 1089 sheep sells 56 of them”

print(“The farmer has”,1089 – 56,”sheep left”)

print(“4 children pick 56 flowers each”)

print(“The children each have”,56/4,”flowers”)

### Activity 2.2.4

>>> 5 \* 3 / 6 + 4

6.5

>>> (5 \* 3) / (6 + 4)

1.5

The parentheses control the order in which the numbers are calculated. Anything in parentheses is evaluated first.

The precedence order is: parenthesis (round brackets), exponential, division and multiplication, subtract and add

### Activity 2.2.5

>>> 15 / 2 \* 3 + 2

24.5

To explain the answer:

Multiplication and division have higher precedence and take place first from left to right.

First the division:

7.5 \* 3 + 2

Then the multiplication giving:

22.5 + 2

Then the addition giving:

24.5

### Activity 2.2.6 (Homework)

What is the correct answer to the following expression?

>>> 7 + 4 \* 5

1. 55
2. 27
3. 16
4. 33

Correct solution: b) 27

Multiplication has a higher precedence so 4 \* 5 is calculated first and then 7 is added to 20 to make 27.

What is the correct answer to the following expression?

>>> 6 - 2 / 2 + 5

1. 0.57
2. 10
3. 7
4. 11

Correct solution: b) 10

First 2 is divided by 2 to give 6 - 1 + 5

Then working from left to right 6 – 1 = 5 then 5 + 5 =10

*Hint: If two operators have equal precedence, for example + and -, then the calculations are performed in order from left to right.*

# Week 3

## Lesson 1 solutions

### Activity 3.1.1

SyntaxError: invalid syntax

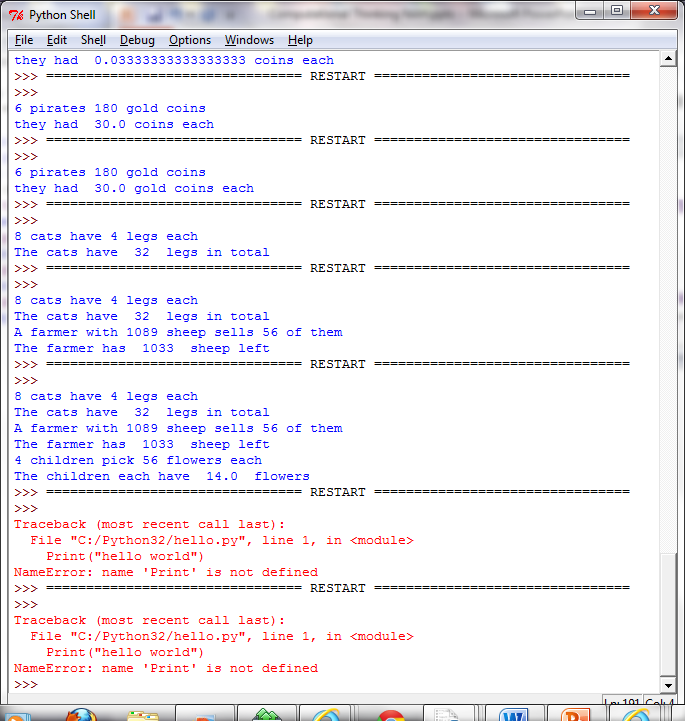
In computer science ‘syntax’ means the set of rules that describes how the language is constructed. A syntax error means that Python does not know what to do because the language used does not follow the correct rules.

The correct code is:

>>>print ("hello world")

### Activity 3.1.2

>>>Print ("hello world")



This is a runtime error. The message says it is a ‘NameError’ and that the name ‘Print’ is not defined. Always read the error messages. They are there to help you.

### Activity 3.1.3

|  |  |
| --- | --- |
| **Type of error** | **Description of the error** |
| TypeError | When an operation is attempted that is invalid for that type of data |
| RuntimeError | An error occurs when the program is running |
| NameError | When a name is used that is not known about (often a misspelt variable name) |
| ZeroDivisionError | Dividing a number by zero |
| KeyBoardInterrupt | When a program is interrupted from the keyboard by pressing control+c |

### Activity 3.1.4

Nothing happens as the # shows that this is a comment statement which the Python program ignores.

*Hint: Always use comments to make your programs readable. Include your name and date and an explanation of what the program does at the beginning of every program.*

### Activity 3.1.6

print("Emily Frances")

print(14)

# Week 3

## Lesson 2 solutions

### Activity 3.2.1

See:

<http://en.wikipedia.org/wiki/George_Boole>

### Activity 3.2.2

>>>type("Fred") string

>>>type (198) integer

>>>type (88.9) float

>>>type(True) boolean

>>>type(alse) boolean

### Activity 3.2.3

|  |  |  |  |
| --- | --- | --- | --- |
| **Expression** | **Predicted data type** | **Type command** | **Result** |
| “hello world” |  | type("hello world") | string |
| False |  | type(False) | boolean |
| 15 |  | type(15) | integer |
| 35.6 |  | type(35.6) | float |
| -999 |  | type(-999) | integer |
| “15” |  | type(“15”) | integer |
| “False” |  | type(“False”) | string |
| True |  | type(True) | boolean |
| 0.001 |  | type(0.001) | float |

### Activity 3.2.4

print("I am ",myName,"I am",myAge,"my eyes are",myEyes,"I am",myHeight,"tall")

*Hint: Use alt p to display previous commands.*

**Activity 3.2.5**

|  |  |  |
| --- | --- | --- |
|  | **Valid or invalid variable name?** | **Reason why not valid** |
| 8HouseNumber = 288 | Invalid | Variable names must begin with a letter |
| houseNumber = 288 | Valid |  |
| house Number = 288 | Invalid | Variable names cannot have spaces in them |
| house\_number = 288 | Valid | “\_” is a valid character in a variable name |
| import = 288 | Invalid | Import is a Python command |

* What type of error do you get when using an invalid variable name?

**SyntaxError: invalid syntax**

### Activity 3.2.6

The program adds the two numbers together and displays the answer.

A program to add three numbers and print the result:

# Programmer Amy Jones 12/8/2013

# adds three numbers

numberOne=15

numberTwo=23

numberThree=76

answer=numberOne + numberTwo + numberThree

print("The answer is ",answer)

### Activity 3.2.7

Data types in Python:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data type** | **Python abbreviation** | **Explanation** | **Example** |
| integer | int | A whole number | 45 |
| string | str | A sequence of characters which can include letters, spaces and other characters | “Have a nice day” |
| float | float | A number with a fractional part | 16.76 |
| boolean | bool | Boolean or logical data can only have one of two values. In Python they are either True or False | True or False |

# Week 4

## Lesson 1 solutions

### Activity 4.1.1

myName=input("Please enter your name: ")

myFood=input("Please enter your favourite food: ")

print("The favourite food of",myName,"is",myFood)

**Note**: the variable name in the program in Activity 4.1.1 is ‘myAnswer’. In this longer program with anther variable a more meaningful name is ‘myName’.

*Hint: Remember the input function* ***always*** *inputs values as a string.*

### Activity 4.1.2

The program cannot add the number 10 to a string, which is what the user is asked to input in this code.

Corrected program:

age=int(input("Please enter your age: "))

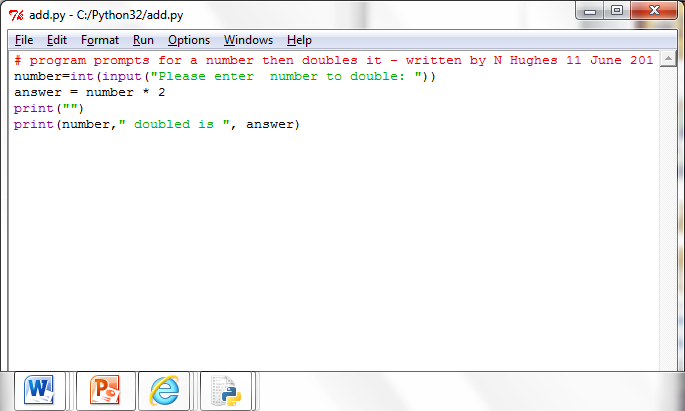
agePlusTen = age + 10

print("You will be",agePlusTen,"in 10 years")

If you want to input real numbers (numbers with a decimal point) use the float function.

*Hint: Use the int function to convert from a string to an integer if you want to input whole numbers.*

### Activity 4.1.3



# Week 4

## Lesson 2 solutions

**Activity 4.2.1**

>>>one="cheese"

>>>two="onion"

My favourite crisps are cheese and onion. I love them!

>>>print("My favourite crisps are {0} and {1}. I love them!".format(one,two))

cheese and onion and cheese and onion and cheese and onion

>>>print("{0} and {1} and {0} and {1} and {0} and {1}".format(one,two))

cheesecheesecheese and oniononiononion

>>>print("{0} {0} {0} and {1} {1} {1}".format(one,two))

You guessed it. The best crisps are onion and ... cheese.

>>>print("You guessed it. The best crisps are {1} and ... {0}.".format(one,two))

*Remember:*

* *The .format method is about formatting strings so the .format fields {0} must be a string.*
* *The format field {0} placeholders are replaced (in order) by the values given in .format brackets.*
* *The values in the .format brackets are separated by commas.*

### Activity 4.2.2

To display 5 decimal places:

number = 765.87641987

print("The answer is {0:.5f}".format(number))

* green is the print command
* yellow is the string
* pink is the field format (place holder)
* grey is the .format method
* blue is the name of the variable. As it is the first variable it is referred to as “0” in the field format.

*Hint: Take care with the format of this command. The zero refers to the field format (placeholder) for the variable “number”. The ‘:’ is a delimiter, the ‘.5’ gives the number of decimal places and the ‘f’ means fixed point.*

*Hint: String formatting is a mini-language in its own right. Take a little time to get familiar with it. You will find it useful later on. If you want to see all the details look at* <http://docs.python.org/2/library/string.html#formatstrings>

### Activity 4.2.3

Program to calculate and display the tip:

MealCost=float(input("Enter total cost of the meal: £"))

TipPercentage=float(input("Enter what percentage to give as a tip (%):"))

Tip = (MealCost\*TipPercentage/100)

print("The tip is £ {0:.2f}".format(Tip))

Program also to calculate total cost of the meal including the tip:

MealCost=float(input("Enter total cost of the meal: £"))

TipPercentage=float(input("Enter what percentage to give as a tip (%):"))

Tip = (MealCost\*TipPercentage/100)

**Extension:**

print("The tip is £ {0:.2f}".format(Tip))

MealPlusTip = Tip + MealCost

print("Total cost of the meal is £ {0:.2f}".format(MealPlusTip))

# Week 5

## Lesson 2 solutions

### Activity 5.2.1

**Example solution**

|  |  |
| --- | --- |
| **Relation statement** | **Operator** |
| Entered password equal to saved password | Equal to |
| People more than 18 years old can vote | Greater than |
| Temperature less than zero degrees water freezes | Less than |
| Cost of tickets for a concert greater than £30 I cannot afford to go | Greater than |
| Temperature in the room greater than 23 degrees – open the window | Greater than |
| Exam result greater than or equal to 75% get an A | Greater than or equal to |

### Activity 5.2.2

|  |  |  |  |
| --- | --- | --- | --- |
| **Relational operator** | **Operator** | **Example** | **Evaluates to** |
| Equal to | == | “fred” == “sid” | False |
| Not equal to | != | 8 != 8 | False |
| Greater than | > | 10 > 2 | True |
| Greater than or equal to | >= | 5 >= 5 | True |
| Less than | < | 40 < 34 | False |
| Less than or equal to | <= | 2 < = 109 | True |

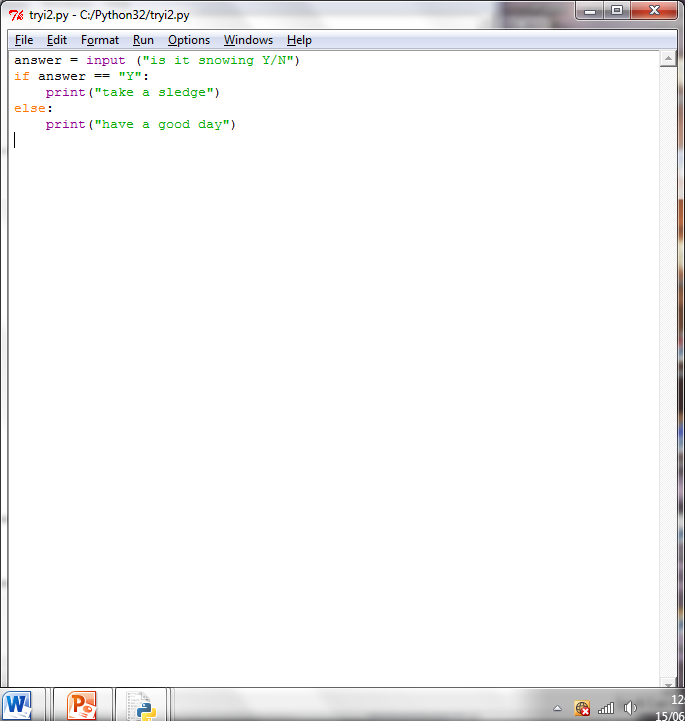
You may have already come across these operators as inequalities in Maths.

### Activity 5.2.3

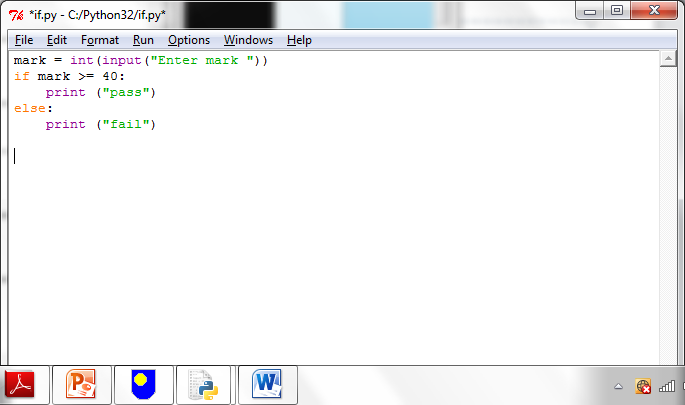
You may find it useful to ask ‘Is “Fred” equal to “Sid”?’ or ‘Is 8 not equal to 8?’ or ‘Is 10 greater than 1’ and answer yes (true) or no (false).

*Hint: A single equal sign “=” is used to assign a value to a variable; a double equal sign “==” is used to test that two values are equal.*

### Activity 5.2.4



### Activity 5.2.5



### Activity 5.2.6

# check if it is safe to cross the road or not

colour=input("Is it a green or red man?")

if colour == "green":

print("It is safe for you to cross.")

else:

print("STOP! It is not safe to cross.")

Hint: In Python white space has a meaning. Code that is at the same position from the margin is grouped into blocks. Python expects blocks to be indented by 4 spaces.

The commands format/indent and format/dedent allow you to indent and dedent highlighted blocks of code.

### Activity 5.2.7

# password checking

password=input("Please enter password: ")

newPassword=input("Please re-enter password: ")

if password == newPassword:

print("Access granted")

else:

print("Access denied")

# Week 6

**Lesson 1 solutions**

### Activity 6.1.2

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Mouse | Computer program |
| Microphone | Web browser |
| CPU | Windows operating system |
| Hard disk drive | Word processor |
| Digital camera | Spreadsheet |
| DVD drive | Mobile phone app |
| Touch screen | Android operating system |
| Speakers |  |
| Printer |  |
| SD memory card |  |

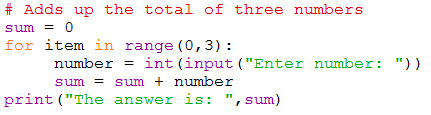
### Activity 6.1.3 (homework)

|  |  |
| --- | --- |
| What is a digital computer? | A digital computer is a programmable data processor:  Digital means the computer only understands binary values 0 and 1.  Programmable means the computer follows instructions stored in a program.  Data processor means the computer does things with data. |
| Define computer hardware. | Hardware is the physical components that make up a computer. |
| Give an example of computer hardware. | CPU |
| Define computer software. | Software is the programs that can be run on a computer. |
| Give an example of computer software. | Operating system |

# Week 6

## Lesson 2 solutions

### Activity 6.2.1

****A program to add three numbers input by the user:

Note: this small program redefines the built-in function sum. This is fortunately not an issue in this small program but could be in a different situation so is not good practice.

Explanation of program using the input-process-output model:

INPUT

User inputs characters using keyboard

PROCESS

Program executes

OUTPUT

Characters displayed on screen

### Activity 6.2.2

# a random number is given by the randint() function

import random

answer= random.randint(1,6)

if answer == 1:

print("You will make a new friend this week")

elif answer == 2:

print("You will do well in your GCSEs")

elif answer == 3:

print("You will find something you thought you’d lost")

elif answer == 4:

print("You will make an important decision today")

elif answer == 5:

print("You will have to ask someone for help this week")

else:

print("You will have to apologise to someone for something")

Make sure you run the program a few times to check it is working.

### Activity 6.2.3 (homework)

Answer will vary depending on pupil’s choice of action. If pupil selects running a mobile phone app:

INPUT

User selects icon on touch screen

PROCESS

App program executes

OUTPUT

Images/sound/text displayed on screen

# Week 7

## Lesson 1 solutions

### Activity 7.1.1

#### #Snippet 1



|  |  |
| --- | --- |
| Outcome: | Traceback (most recent call last):  File "C:/Documents and Settings/jodill/Desktop/27.1.1.py", line 3, in <module>  print(Number1 \* Number2)  TypeError: can't multiply sequence by non-int of type 'str' |
| Why this happened: | The program cannot multiply one variable by the other because both variables are strings. |

#### #Snippet 2



|  |  |
| --- | --- |
| Outcome: | The program prints Number1 as many times as Number2. So, if Number1 = 3 and Number2 = 4, the outcome is:  Enter your first number:3  Enter your second number:4  3333 |
| Why this happened: | Number1 is a string whereas Number2 is an integer. So the program prints the string the number of times specified by the integer. |

#### #Snippet 3



|  |  |
| --- | --- |
| Outcome: | Number1 multiplied by Number2. So, if Number1 = 4 and Number2 = 5, the outcome is:  Enter your first number:4  Enter your second number:5  20 |
| Why this happened: | Both variables are integers, so the program multiplies them together and prints the answer. |

#### #Snippet 4



|  |  |
| --- | --- |
| Outcome: | The program prints a float. So, if Number1 = 3 and Number2 = 5, the outcome is:  Enter your first number:3  Enter your second number:5  15.0 |
| Why this happened: | The program converts both inputs to floats so when the program multiplies these together and prints the answer, the answer is also a float. |

#### #Snippet 5



|  |  |
| --- | --- |
| Outcome: | The program prints a float. So, if Number1 = 2 and Number2 = 4, the outcome is:  Enter your first number:2  Enter your second number:4  8.0 |
| Why this happened: | The program multiplies a float by an integer, so the answer is a float. |

**#Snippet 6**



|  |  |
| --- | --- |
| Outcome: | Traceback (most recent call last):  File "C:/Documents and Settings/jodill/Desktop/27.1.1.py", line 3, in <module>  print(Number1 \* Number2)  TypeError: can't multiply sequence by non-int of type 'float' |
| Why this happened: | The program cannot multiply a string by a float. |

### Activity 7.1.2

|  |  |
| --- | --- |
| **Binary** | **Denary** |
| 0001 | 1 |
| 0010 | 2 |
| 0101 | 5 |
| 1000 | 8 |
| 1001 | 9 |
| 1100 | 12 |
| 1011 | 11 |
| 1111 | 15 |

|  |  |
| --- | --- |
| **Denary** | **Binary** |
| 4 | 0100 |
| 6 | 0110 |
| 10 | 1010 |
| 0 | 0000 |
| 3 | 0011 |
| 7 | 0111 |
| 14 | 1110 |
| 13 | 1101 |

|  |  |
| --- | --- |
| What range of numbers can be represented by 8 bits? | 0–255 |

|  |  |
| --- | --- |
| Underline the most significant bit in this binary number. | 1011 0011 |
| What weighting does it have? | 27 |
| What is its value in denary? | 128 |

**Activity 7.1.3**

|  |  |
| --- | --- |
| How many days in a week? | 0111 |
| How many months in a year? | 1100 |
| How many fingers (including the thumb) on one hand? | 0101 |
| How many toes on two feet? | 1010 |
| How many lives does a cat have? | 1001 |

# Week 7

## Lesson 2 solutions

### Activity 7.2.1

def display\_results(score):

# Function that displays a message depending on value of score

# Written by Amy Jones 31 June 2013

if score < 50:

print("You have lost")

else:

print("You have won")

# Week 8

## Lesson 1 solutions

### Activity 8.1.1

|  |  |
| --- | --- |
| **Binary** | **Denary** |
| 0101 | 5 |
| 1010 | 10 |
| 0100 | 4 |
| 1000 | 8 |
| 0011 | 3 |
| 0001 | 1 |
| 1111 | 15 |
| 0010 | 2 |

|  |  |
| --- | --- |
| * What are these denary numbers in binary? | |
| **Denary** | **Binary** |
| 7 | 0111 |
| 3 | 0011 |
| 10 | 1010 |
| 0 | 0000 |
| 9 | 1001 |
| 12 | 1100 |
| 6 | 0110 |
| 13 | 1101 |

### Activity 8.1.2

|  |  |
| --- | --- |
| How many states does a transistor have? | 2 |
| How many bits in a nibble? | 4 |
| What range of unsigned integer numbers can be represented by 8 bits? | 0–255 |
| What is the position of the most significant bit in a byte? | 8 (far left) |
| How many different values can be represented with 6 bits? | 64 (in the range 0–63) |

### Activity 8.1.3

|  |  |
| --- | --- |
| What comes next? | |
| 1111 | 1 0000 |
| 0011 1111 | 0100 0000 |

|  |  |
| --- | --- |
| What comes before? | |
| 1000 | 0111 |
| 1111 | 1110 |

### Activity 8.1.4

|  |  |
| --- | --- |
| * What are these binary numbers in denary? | |
| **Binary** | **Denary** |
| 00001011 | 11 |
| 11010010 | 210 |
| 01010000 | 80 |
| 01001000 | 72 |
| 10010111 | 151 |
| 11001100 | 206 |
| 10101010 | 170 |
| 11111111 | 255 |

|  |  |
| --- | --- |
| * What are these denary numbers in binary? | |
| **Denary** | **Binary** |
| 65 | 01000001 |
| 100 | 01100100 |
| 78 | 01001110 |
| 29 | 00011101 |
| 223 | 11011111 |
| 80 | 01010000 |
| 14 | 00001110 |
| 55 | 00110111 |

### Activity 8.1.5

A program that inputs a 4 bit binary number and converts it to an unsigned integer.

#Program to convert 4-bit binary numbers to denary

binary\_number = input('Please enter a 4-bit binary number: ')

print((int(binary\_number[3])) + (int(binary\_number[2])\*2) + (int(binary\_number[1])\*4) + (int(binary\_number[0])\*8))

Produces:

>>>

Please enter a 4-bit binary number: 1101

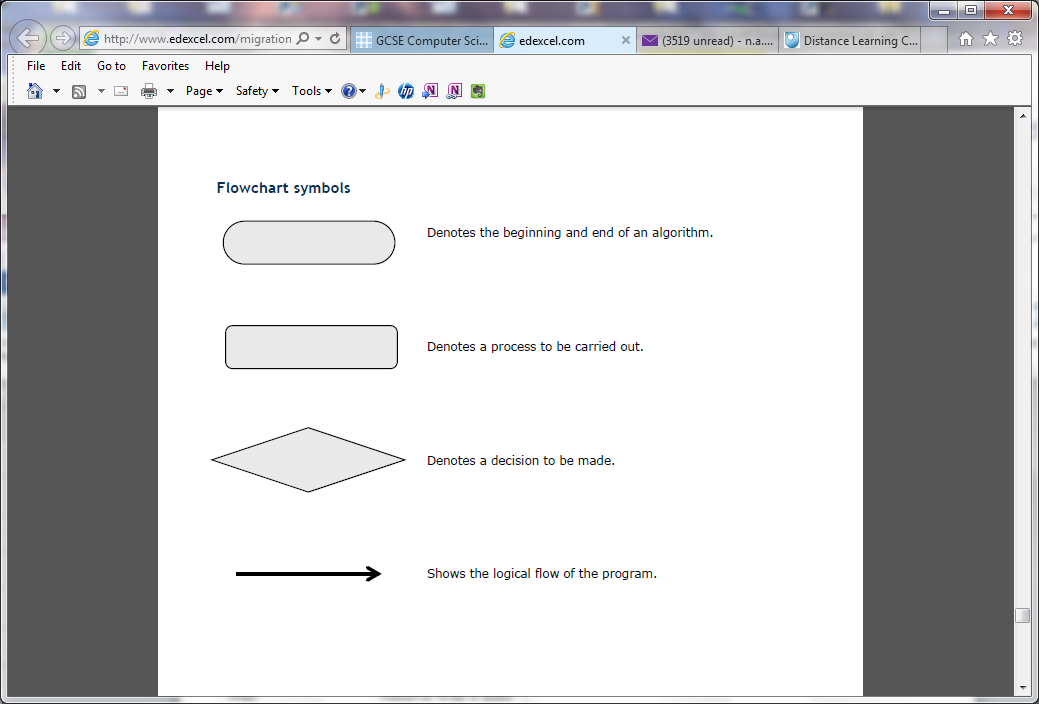
13

>>>

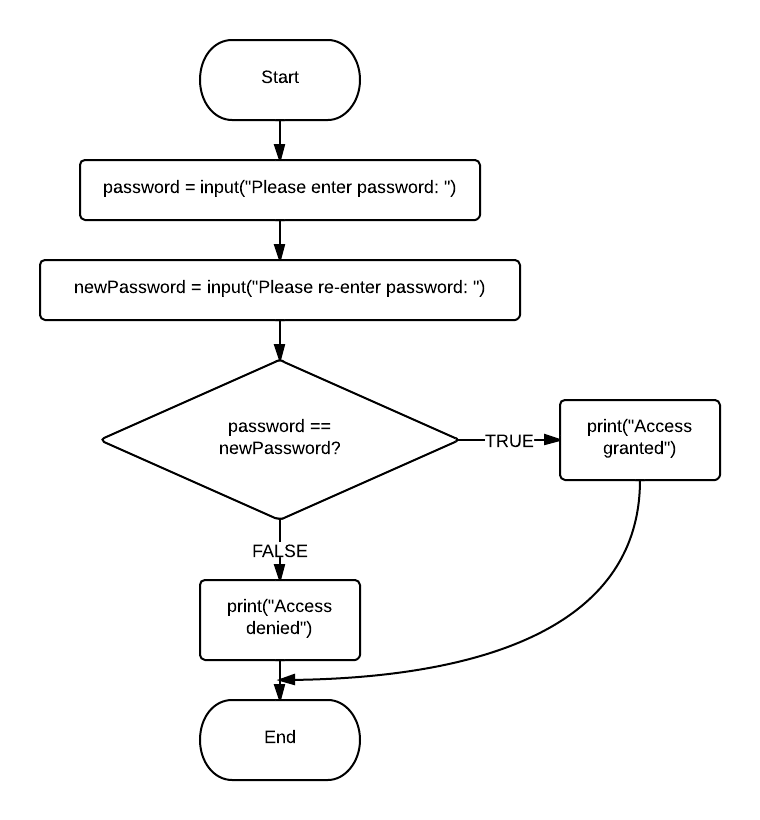
# Week 8

## Lesson 2 solutions

### Activity 8.2.1



### Activity 8.2.2



### Activity 8.2.3

myName="Fred"

myAge=15

print(myName,myAge)

### Activity 8.2.4

Depending on your age, it tells you whether or not you can watch a 12A film:

Age=int(input("enter age"))

if Age >= 12:

print("You can view a 12A film")

else:

print("Sorry, you cannot view a 12A film")

# Week 9

## Lesson 1 solutions

### Activity 9.1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0111 7 | 7 | 0101 | 5 | 1111 15 | 15 |
| 0100 + | 4 + | 0110 + | 6 + | 0111 + | 7 + |
| 1011 | 11 | 1011 | 11 | 10110 | 22 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0011 | 3 | 1011 11 | 11 | 0111 | 7 |
| 0011 + | 3 + | 1111 + 15 + | 15 + | 0010 + | 2 + |
| 0110 | 6 | 11010 26 | 26 | 1001 | 9 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1110 | 14 | 01111 | 15 | 01110 | 14 |
| 0111 + | 7 + | 10101 + | 21 + | 10111 + | 23 + |
| 10101 | 21 | 100100 | 36 | 100101 | 37 |

|  |  |  |  |
| --- | --- | --- | --- |
| 10101010 | 170 | 10101100 172 | 172 |
| 00110010 + | 50 + | 00010010 18 + | 18 + |
| 11011100 | 220 | 10111110 190 | 190 |

**Activity 9.1.2**

|  |  |
| --- | --- |
|  | **Arithmetic shift 1 Left** |
| 00101100 | 01011000 |
| 10011100 | 00111000 |
| 00101000 | 01010000 |
| 00100111 | 01001110 |

|  |  |
| --- | --- |
|  | **Arithmetic shift 2 Right** |
| 00101100 | 00001011 |
| 10011100 | 11100111 |
| 00101000 | 00001010 |
| 00100111 | 00001001 |

|  |  |
| --- | --- |
|  | **Logical shift 1 Right** |
| 00101100 | 00010110 |
| 10011100 | 01001110 |
| 00101000 | 00010100 |
| 00100111 | 00010011 |

|  |  |
| --- | --- |
|  | **Logical shift 1 Left** |
| 00101100 | 01011000 |
| 10011100 | 00111000 |
| 00101000 | 01010000 |
| 00100111 | 01001110 |

**Activity 9.1.3 (homework)**

A program that inputs a binary number of any length and converts to denary.

#Program to convert a binary number of any length into denary

binary\_number = input ('Please enter a binary number: ')

power = len(binary\_number) - 1

value = 0

for digit in binary\_number:

if digit == '1':

value = value + (2\*\*power)

power = power - 1

print(value)

Produces:

>>>

Please enter a binary number: 11111111

255

>>>

# Week 9

## Lesson 2 solutions

### Activity 9.2.1

|  |  |
| --- | --- |
| **Symbol** | **Description** |
| AND | Returns true if both conditions are true. |
| OR | Returns true if any of the conditions are true. |
| NOT | Reverses the outcome of the expression; true becomes false, false becomes true. |

**Activity 9.2.2**

|  |  |  |
| --- | --- | --- |
| **Condition** | **Your answer (true or false)** | **Result** |
| (78 == 10) **or** (6 == 7) |  | False – as neither of the conditions are true |
| (78 == 10) **or** (6 == 6) |  | True – one of the conditions is true (6 == 6) |
| (78 == 10) **and** (6 == 6) |  | False – only one of the conditions is true and both need to be true for an AND |
| (1 < 10) **and** (2 < 10) |  | True – as both conditions are true |
| (1 < 10) **or** (2 < 10) |  | True – as at least one of the conditions is true |
| not ( 5 == 5) |  | False – as the answer to the condition is true as 5 is equal to 5 |
| not (6 < 4) |  | True – as the answer to the condition is false 6 is less than 4 |

### Activity 9.2.3

False

50 is less than 40 so the first condition is false

50 is not greater than 80 so the second condition is false

### Activity 9.2.4

**Truth table showing true and false AND conditions**

|  |  |  |
| --- | --- | --- |
| **Condition 1** | **Condition 2** | **Output** |
| false | false | false |
| true | false | false |
| false | true | false |
| true | true | true |

In AND logical operators, **both** conditions have to be true for the output to be true.

**Truth table showing true and false OR conditions**

|  |  |  |
| --- | --- | --- |
| **Condition 1** | **Condition 2** | **Output** |
| false | false | false |
| true | false | true |
| false | true | true |
| true | true | true |

In OR logical operators, only one of the inputs has to be true for the output to be true.

**Truth table showing true and false NOT conditions**

|  |  |
| --- | --- |
| **Condition 1** | **Output** |
| false | true |
| true | false |
| false | true |
| true | false |

### Activity 9.2.5

# put year groups into key stages

YearGroup = int(input("Enter year group: "))

if YearGroup == 1 or YearGroup == 2:

KeyStage = 1

elif YearGroup >= 3 and YearGroup<= 6:

KeyStage = 2

elif YearGroup >= 7 and YearGroup<= 9:

KeyStage = 3

elif YearGroup == 10 or YearGroup == 11:

KeyStage = 4

print("You are in key stage",KeyStage)

Possible solution to extension activity:

# put year groups into key stages version2

YearGroup = int(input ("What year group are you interested in?"))

if YearGroup == 1 or YearGroup == 2:

print("Key Stage 1")

elif YearGroup >2 and YearGroup <7:

print("Key Stage 2")

elif YearGroup >6 and YearGroup <10:

print("Key Stage 3")

elif YearGroup >=10 and YearGroup <12:

print("Key Stage 4")

elif YearGroup >=12:

print("Sixth form")

### Activity 9.2.6

**Python commands colour coding**

|  |  |  |
| --- | --- | --- |
| **Colour coding** | **What does it show** | **Example** |
| green | string | “hello” |
| purple | function | print() |
| black | variables and data | myName |
| orange | key commands | if |
| red | comment | # This is a comment |

### Week 10 Activity 10.1.2

## Lesson 1 activities

### Activity 10.1.1

|  |  |
| --- | --- |
| **Sign and magnitude binary number** | **Denary** |
| 1100 1101 | -(64+8+4+1) = -77 |
| 0001 1111 | +(16+8+4+2+1) = +31 |
| 1000 1010 | -(8+2) = -10 |
| 0101 1100 | +(64+16+8+4) = +92 |
| 1000 0000 | 0 |
| 1111 1111 | -(64+32+16+8+4+2+1) = -127 |
| 0111 1111 | +(64+32+16+8+4+2+1) = +127 |

### Activity 10.1.2

|  |  |
| --- | --- |
| **Two’s complement binary number** | **Denary** |
| 1100 1101 | -128+64+8+4+1 = 51 |
| 0001 1111 | 16+8+4+2+1 = 31 |
| 1000 1010 | -128+8+2 = -118 |
| 0101 1100 | 64+16+8+4 = 92 |
| 1000 0000 | -128 |
| 1111 1111 | -128+64+32+16+8+4+2+1 = -1 |
| 0111 1111 | 64+32+16+8+4+2+1 = 127 |

### Activity 10.2.1

The program asks for the user’s name and age, adds 10 on to the age and displays the information on the screen.

Python code:

name = input("Enter your name: ")

age = int(input("Enter your age: "))

ageInTen = age + 10

print(name, "will be ",ageInTen,"in 10 years’ time")

### Activity 10.2.2

If the score is less than 50 it displays “you have lost”. If the score is greater than 50 it displays “you have won”.

score = 119

if score < 50:

print("You have lost")

else:

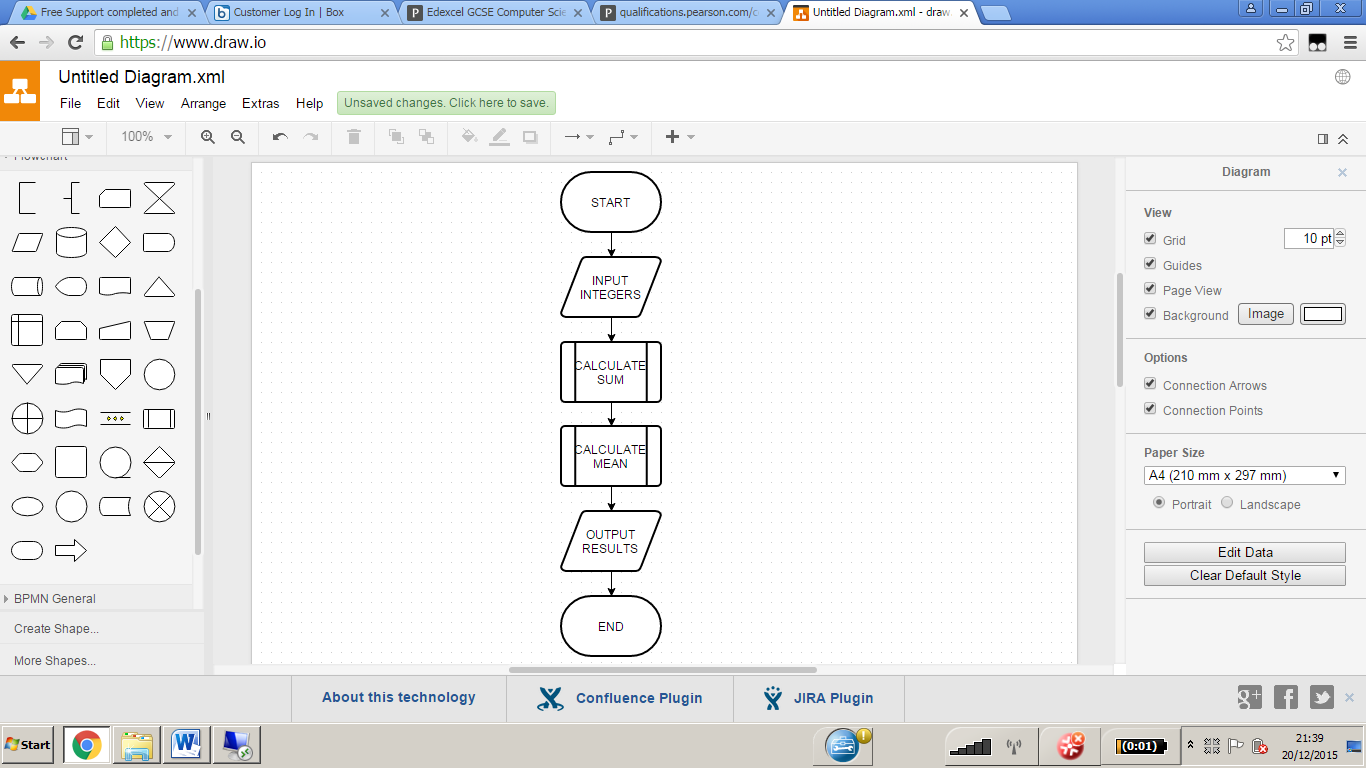
print("You have won")

### Activity 10.2.3

Your task is to create a program to ask a user to input 3 integer numbers, calculate the total and the average of the number and display the results.

* Create a flowchart for this task.
* Write pseudo-code for the same task.
* Write the program.

### Flowchart



### Pseudocode

input num1, num2, num3

total = num1 + num2 + num3

ave = total/3

print(“Total:”,total)

print(“Average:”,ave)

### Python NB no input validation

n1 = int(input(“Enter number 1: “))

n2 = int(input(“Enter number 2: “))

n3 = int(input(“Enter number 3: “))

total = n1 + n2 + n3

ave = total / 3

print(“Total is”,total)

print(“Average is”,ave)

# Week 11

## Lesson 1 solutions

### Activity 11.1.1

|  |  |
| --- | --- |
| **Hexadecimal** | **Binary** |
| 3B | 3B  3 B  0011 1011  00111011 |
| 26 | 26  2 6  0010 0110  00100110 |
| FA | FA  F A  1111 1010  11111010 |
| 61 | 61  6 1  0110 0001  01100001 |
| AA | AA  A A  1010 1010  10101010 |
| 8C | 8C  8 C  1000 1100  10001100 |

### Activity 11.1.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | | | Binary | Hexadecmial | | 10101010 | 10101010  1010 1010  A A  AA | | 01001111 | 01001111  0100 1111  4 F  4F | | 10000001 | 10000001  1000 0001  8 1  81 | | 01110001 | 01110001  0111 0001  7 1  71 | | 11110000 | 11110000  1111 0000  F 0  F0 | | 01001100 | 01001100  0100 1100  4 C  4C | | 01111111 | 01111111  0111 1111  7 F  7F | |

# Week 11

## Lesson 2 solutions

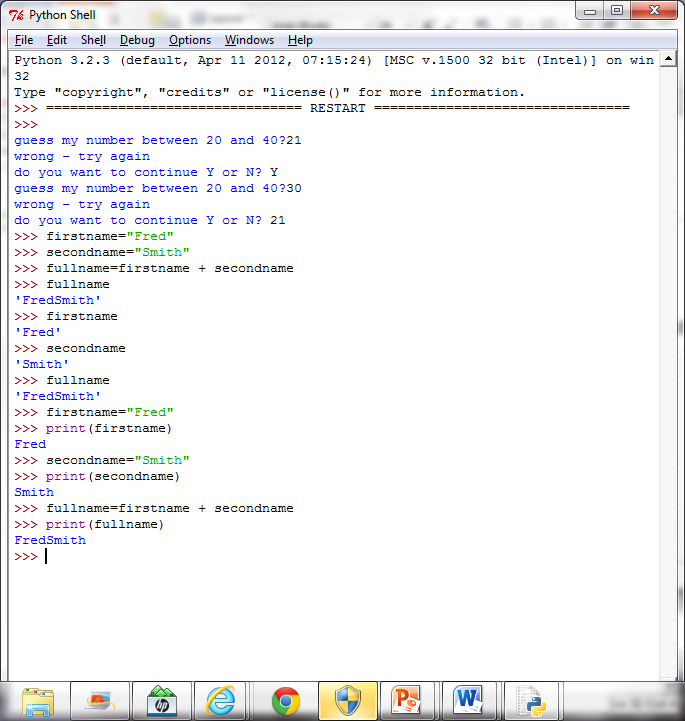
### Activity 11.2.1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | e | l | l | o |  | w | o | r | l | d | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

* The character at index 3 is: l
* The character at index 5 is: ‘ ’ (a space)
* The character at index 0 is: H
* The character at index 11 is: !
* There are 12 characters in this string.

*Hint: Remember that the string indexes always start at zero.*

### Activity 11.2.2



*Hint: Remember that the + symbol allows concatenation of strings.*

### Activity 11.2.3

>>>word[2:5]

'ZZA'

>>>word[1:2]

'I'

>>>word[0:3]

'PIZ'

>>>word[0:5]

'PIZZA'

### Activity 11.2.4

myVariable = "watch #BBCClick today"

To extract ‘#BBCClick’:

>>>myVariable[6:15]

To extract ‘watch’:

>>>myVariable[0:5]

To extract ‘today’:

>>>myVariable[16:21]

# Week 12

## Lesson 1 solutions

### Activity 12.1.1

First programmer – Ada Lovelace. There are many facts do be found online.

### Activity 12.1.2

* Column 4 does not attempt to list all examples of ‘something special’.

|  |  |  |  |
| --- | --- | --- | --- |
| **High level programming language** | **When was it created?** | **Who invented it?** | **Find out something special about the language (e.g. is it used for a particular purpose?)** |
| FORTRAN | Early 1950s. First compiler delivered 1957 | John Backus (at IBM) | Numeric, scientific use. |
| Java | 1995 | Sun Microsystems | Cross-platform |
| C | 1978 | Dennis M. Ritchie (at Bell Labs) | Originally used to develop the operating system UNIX |
| Visual Basic  (now superseded by VB.Net) | 1991 | Microsoft | Event-driven with an IDE |
| PHP | 1994 | Rasmus Lerdorf | Server-side scripting originally used for web development |
| Python | Implementation started in 1989. | Guido van Rossum (at CWI) | General purpose language |
| Ruby | 1993 | Yukihiro Matsumoto | General purpose, object-oriented |

# Week 12

## Lesson 2 activities

### Activity 12.2.1

|  |  |
| --- | --- |
| **Things to do with lists** | **Commands** |
| Create a list | Mylist=[“string”,10,78.45] |
| Reference an item in a list | |  |  |  |  |  | | --- | --- | --- | --- | --- | | apples | oranges | lemon | pear | lime | | 0 | 1 | 2 | 3 | 4 |   **[start index: end index]**  Start index is the index to start at (remember that indexing starts at zero).  End position is the index AFTER the index required.  [9] refers to that particular item in the list  [-2] refers to index counting from the end |
| Delete an item in a list | del Listname[index] |
| Append an item to the end of a list | Listname.append(item) |

### Activity 12.2.2

>>>mylist[1]

'orange'

>>>mylist[1:3]

'orange', 'lemon'

>>>mylist[-1]

'lime'

*Hint: Negative indexes display items from the end of the list.*

The command that just displays ‘apple’ is:

>>>mylist[0]

The command that displays ‘lemon’ and ‘pear’ is:

>>>mylist[2:4]

To display the whole list:

>>>mylist=["chips","pizza","chicken","frozen peas","cherries"]

To display the item at index position 3:

>>>mylist[3]

'frozen peas'

To display the item at index position 0:

>>>mylist[0]

'chips'

To display the items at index position 1 to 4:

>>>mylist[1:4]

'pizza', 'chicken', 'frozen peas'

### Activity 12.2.3

Changing Amy’s mark for English from 67 to 72:

>>>Marks[3]=72

>>>Marks

['Amy', 'Jones', 'English', 72, 'Maths', 76, 'Computer Science', 96]

Adding Amy’s mark for Physics:

>>>Marks.append("Physics")

>>>Marks.append(65)

>>>Marks

['Amy', 'Jones', 'English', 72, 'Maths', 76, 'Computer Science', 96, 'Physics', 65]

Removing Maths and the Maths score from the list:

>>>del Marks[4:6]

>>>Marks

['Amy', 'Jones', 'English', 72, 'Computer Science', 96, 'Physics', 65]

A program to find the average score for the three subjects:

>>>average = (Marks[3] + Marks[5] + Marks[7]) / 3)

>>>average

77.66666666666667

# Week 13

## Lesson 1 solutions

### Activity 13.1.1



**MOV R1, #50**

**MOV R2, #60**

**ADDS R0, R1, R2**

**MOV R7, #1**

**1111000011110000**

**1011100011100010**

**1010010010010100**

Python (high level language)

Assembly language

Machine code

Compiler

Assembler

### Activity 13.1.2

* Write a mock translator program using the data in the table below.
  + Prompt the user to choose a high level or assembly language, and translate to the next lower level and display the result.

|  |  |  |
| --- | --- | --- |
| **High level language command** | **Assembly language** | **Machine code** |
| If x == y | CMP | 10110011 |
| Total = num1 + num2 | ADD | 10000101 |
| Minus = num2 – num1 | SUB | 10010000 |

* **Extension (homework):** enter a high level language command and translate it into machine code in two stages.

# # Mock translator

# highlevel = "if x == y:\n\ttotal = num1 + num2\n\tminus = num2 – num1\n"

# assembly = "CMP\nADD\nSUB\n"

# machinecode = "10110011\n10000101\n10010000\n"

# level = ""

# while level not in ("high","low"):

# level = input("Which level of language to start with [high/low]: ").lower()

# print()

# print("Source code\n===========")

# if level == "high":

# print(highlevel)

# print("Translating to lower level language: assembly language.")

# print("=======================================================")

# print(assembly)

# elif level == "low":

# print(assembly)

# print("Translating to lower level language: machine code.")

# print("==================================================")

# print(machinecode)

# EXTENSION

# # Mock translator extension

# highlevel = ["if x == y:","total = num1 + num2","minus = num2 – num1"]

# assembly = ["CMP","ADD","SUB"]

# machinecode = ["10110011","10000101","10010000"]

# for i in range(len(highlevel)):

# print(str(i+1)+".",highlevel[i])

# print("Choose a high level language instruction [1/2/3]: ")

# instruction = 0

# while instruction not in ('1','2','3'):

# instruction = input()

# instruction = int(instruction)

# print("Translating to lower level language: assembly language.")

# print("=======================================================")

# print(assembly[instruction-1])

# print("Translating to lower level language: machine code.")

# print("==================================================")

# print(machinecode[instruction-1])

# Week 13

## Lesson 2 activities

### Activity 13.2.1

for number in range(10):

print("I like Ellie Goulding")

**Activity 13.2.2**

The for loop iterates over the range from 0 to 9 each time assigning the value to the variable number.

Each time through the loop the print command is executed and the current value assigned to number is printed on the screen.

The program displays the numbers from 0 to 9 on the screen one line at a time.

A program that prints the numbers from 1 to 15:

for number in range(15):

print(number)

A program that prints the numbers from 1 to 8, with the square of each number, both on the same line:

for number in range(1,8):

print(number,number\*number)

A program that prints out the 9 times table (up to 20 x 9):

for number in range(1,21):

print(number,number\*9)

A program that asks the user which times table they want and then displays that table:

table=int(input("What times table would you like? "))

for number in range(1,21):

print(number,number\*table)

A program that asks for a start value and end value and then counts in fives between those numbers:

start=int(input("Start number "))

end=int(input("End number "))

for number in range(start,end,5):

print(number)

### Activity 13.2.3

* The ‘name’ variable will be assigned to each item in the list each time through the for loop.
* A program that creates a list of things you would take to a desert island and displays each item a line at a time:

mylist=["cat","book","music","swimming costume"]

for item in mylist:

print("I would take on my desert island a",item)

### Activity 13.2.4

myList=["cat","dog","cow","donkey","rabbit","canary"]

for next in myList:

if next[0] == "c" :

print(next)

*Hint: Remember to indent the program blocks after each ‘:’*

# Week 14

## Lesson 1 solutions

### Activity 14.1.1

|  |  |
| --- | --- |
| **Function** | **Hardware component** |
| A temporary storage area for data and program instructions while a program is running | RAM (Random Access Memory) |
| A microprocessor which carries out the instructions in computer programs by performing arithmetic and logic operations, and controls inputs and outputs | CPU (or processor) |
| A persistent data storage area for data and program instructions | Secondary storage, e.g. hard disk drive, solid state memory |
| To provide connections to the input and output peripherals such as printers, mouse, keyboard, touch screen, speakers, networks, and so on | Sound card, network card and graphics card |

### Activity 14.1.2

CPU

Data, control and address bus

Input / output devices

Memory

### Activity 14.1.3

|  |  |
| --- | --- |
| **Function** | **Type of bus** |
| Sends and receives signals that control the CPU and other parts of the computer system | Control bus |
| Carries the address of memory locations used to store data and program instructions | Address bus |
| Transfers the binary data around the computer | Data bus |

### Activity 14.1.4



2 USB connectors (mouse and keyboard)

Ethernet network connection

HDMI connector for monitor

CPU processor and RAM memory

Audio

SD card (memory)

Power supply

# Week 14

## Lesson 2 activities

### Activity 14.2.1

* The program finds the mean of the numbers in the list.
* The variable ‘next’ holds the value of each item in the list each time through the loop.
* The variable ‘count’ is used to sum the total values. The variable ‘count’ is initialised to zero before loop starts.
* The function ‘len(myNumbers)’ finds the length of the list (number of elements in the list).
* If you add or append more numbers to the list the program will still calculate the mean of the numbers.
* If a print statement was added to the for loop this is what would be displayed:

1 1

11 12

7 19

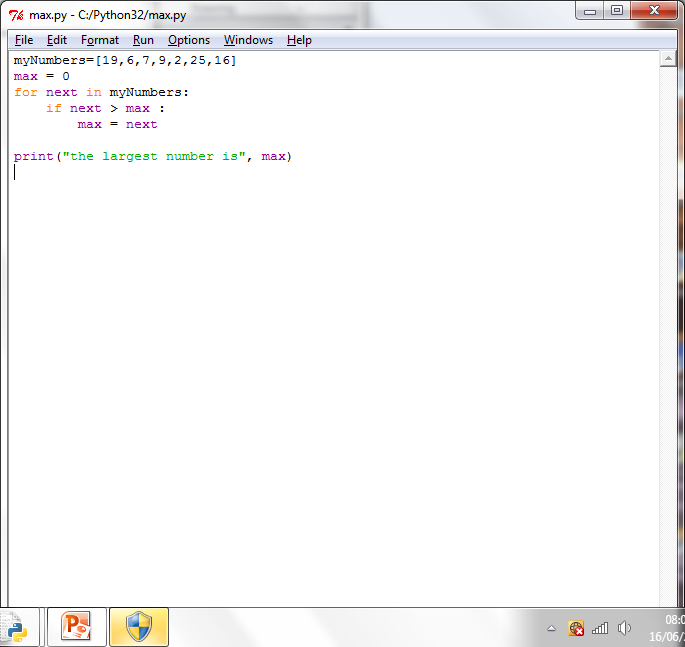
8 27

12 39

14 53

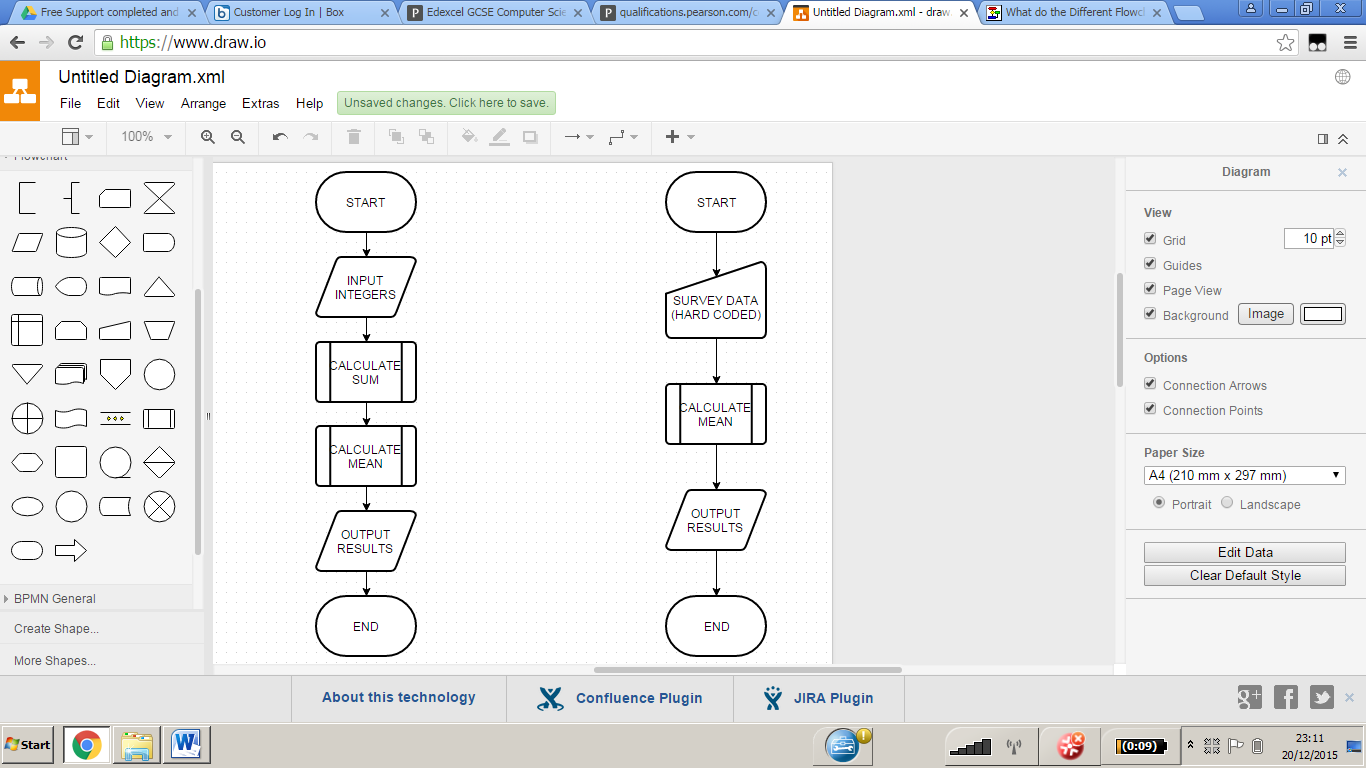
The answer is 8.833333333333334

### Activity 14.2.2



### Activity 14.2.3

* Write a program to calculate the average popularity position for the languages you identified in Activity 12.1.3. First create a flowchart to show how you will solve this problem.



# Language popularity analyser

# Edit the following lines to represent your actual data from lesson 12

languages = ['C', 'Python', 'Java', 'VB']

popSource1 = [1,3,4,2]

popSource2 = [2,1,3,4]

popSource3 = [3,1,2,4]

for i in range(len(languages)):

ave = ( popSource1[i] + popSource2[i] + popSource3[i] ) / 3

print("Mean popularity for",languages[i],"is",ave)

# Week 15

## Lesson 1 activities

### Activity 15.1.2

# Week 15

## Lesson 2 activities

### Activity 15.2.1

* The program keeps asking the user if they are tired until they say ‘Y’ and then it tells them to go to sleep.
* The condition being tested is ‘answer != “Y”’

### Activity 15.2.2

answer="N"

counter=0

while answer != "Y" :

print("Are you hungry? You have been asked {0} times".format(counter))

answer = input("Please respond Y or N:")

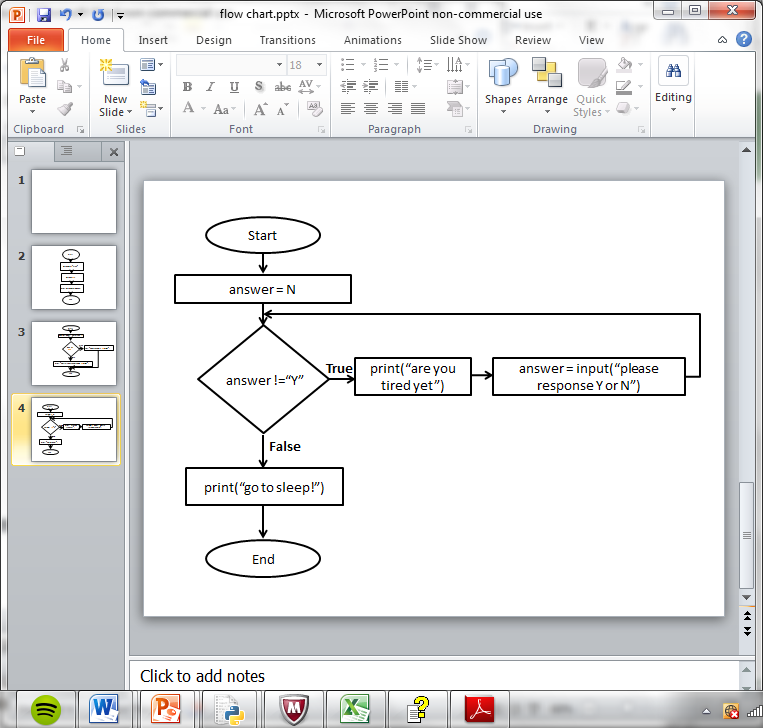
counter=counter + 1

print("Please get something to eat!")

### Activity 15.2.3

* The program prints out the numbers from 1 to 8.
* The ‘sentry’ variable is number.
* The condition is: number < 8

### Activity 15.2.4



# Week 16

## Lesson 2 solutions

### Activity 16.2.1

* The program returns a random number.
* The ‘import random’ command imports into the program additional functions that can then be used.
* If you alter the values to 1 and 100 the range that the random number can be in is between 1 and 100.

### Activity 16.2.2

import random

answer="Y"

while answer != "N":

number = random.randint(1,6)

print("The dice says {0}!".format(number))

answer=input("Do you wish to continue? Y or N:")

print("Good bye... play again soon.")

### Activity 16.2.3

# this program asks the user to guess a number between 1 and 10

import random

RandomNo = random.randint(1,10)

Count=0

while Count<10:

YourGuess = int(input("Enter a number between 1 and 10: "))

if YourGuess == RandomNo:

print("Well done. You guessed correctly.")

break

else:

Count=Count+1

if Count<10:

print("Try again! ")

else:

print("Bad luck. You're out of guesses.")