**LESSON ACTIVITIES FOR YEAR 1, TERM 2**

# Week 1

## Lesson 1 Activities

### Activity 1.1.1

* What is a computer network?

### Activity 1.1.2

* How many computer networks have you used today?

|  |  |
| --- | --- |
| Network | Used today?  (Tick as appropriate) |
| School computer network |  |
| GSM network |
| 3G / 4G network |
| Bluetooth network |
| WiFi network |
| Home network |
| Internet |

### Activity 1.1.3

* Write a list of the activities computer networks make possible.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## Lesson 2 Activities

### Activity 1.2.1

Lists and for loops revisited

* What does this command do?

>>>countries= ["Japan","Germany","USA","China","Austria","Turkey","Mexico"]

* How would you display all the values in countries? Print(countries)
* How would you display ‘China’? print(countries[3])
* How would you display ‘Japan’ print(countries[0])

*Hint: Remember that indexes for lists always start at 0.*

* How would you address the third item in the list? Countries[2]
* Copy and run this program. Explain how it works.

countries= ["Japan","Germany","USA","China","Austria","Turkey","Mexico"]

for name in countries:

print("This is one of my favourite countries ", name)

The program repeats the phrase “This is one of my favourite countries” + the country’s name for every country in the list

* Copy and run this program, which uses list comprehension. Explain how it works.

length=10

myList= [43 for number in range(length)]

print(myList)

* Write a program that creates a list with each element initialised to 0?

### Activity 1.2.2

**Battleships: a game using two-dimensional arrays (lists)**

How to set up the game

* Each player decides at which index locations in the two-dimensional array [row, column] to place their ships.
* They each have five ships:

1. A battle ship that takes up five index spaces
2. A cruiser that takes up four index spaces
3. A submarine that takes up three index spaces
4. A destroyer that takes up two index space
5. Four spy ships, disguised as fishing boats, that each take up one index space.

* None of the ships may be placed diagonally; they must all be placed in straight lines either horizontally or vertically. It is legal (but not required) for two or more ships to be next to each other. The ships are marked by blocking in the appropriate spaces.

How to play the game

* Players take turns taking shots at each other’s ships. A shot is taken by calling out the index locations on the 8 x 8 two-dimensional array. The array index locations are given [row, column] e.g. [2, 6].
* Each player takes one shot at a time.
* If the player calls the coordinates of an index location where a ship is located, their opponent tells them so by saying ‘hit’. If they miss, their opponent says ‘miss’.
* Players mark the shots they take on their ‘Opponent’ array, and whether each shot was a hit or a miss to keep track of their shots. Players may also mark the ‘Self’ array to show the shots taken by their opponent.
* A ship is sunk when all of its index locations have been hit. When this happens, the player whose ship was sunk says, for example, ‘You sank my spy ship.’
* The winner is the player who manages to sink all their opponent’s ships.

## Battleships: My battle ships

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [0,0] | [0,1] | [0,2] | [0,3] | [0,4] | [0,5] | [0,6] | [0,7] |
| [1,0] | [1,1] | [1,2] | [1,3] | [1,4] | [1,5] | [1,6] | [1,7] |
| [2,0] | [2,1] | [2,2] | [2,3] | [2,4] | [2,5] | [2,6] | [2,7] |
| [3,0] | [3,1] | [3,2] | [3,3] | [3,4] | [3,5] | [3,6] | [3,7] |
| [4,0] | [4,1] | [4,2] | [4,3] | [4,4] | [4,5] | [4,6] | [4,7] |
| [5,0] | [5,1] | [5,2] | [5,3] | [5,4] | [4,5] | [5,6] | [5,7] |
| [6,0] | [6,1] | [6,2] | [6,3] | [6,4] | [6,5] | [6,6] | [6,7] |
| [7,0] | [7,1] | [7,2] | [7,3] | [7,4] | [7,5] | [7,6] | [7,7] |
| [8,0] | [8,1] | [8,2] | [8,3] | [8,4] | [8,5] | [8,6] | [8,7] |

## Battleships: My opponent’s battle ships

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [0,0] | [0,1] | [0,2] | [0,3] | [0,4] | [0,5] | [0,6] | [0,7] |
| [1,0] | [1,1] | [1,2] | [1,3] | [1,4] | [1,5] | [1,6] | [1,7] |
| [2,0] | [2,1] | [2,2] | [2,3] | [2,4] | [2,5] | [2,6] | [2,7] |
| [3,0] | [3,1] | [3,2] | [3,3] | [3,4] | [3,5] | [3,6] | [3,7] |
| [4,0] | [4,1] | [4,2] | [4,3] | [4,4] | [4,5] | [4,6] | [4,7] |
| [5,0] | [5,1] | [5,2] | [5,3] | [5,4] | [4,5] | [5,6] | [5,7] |
| [6,0] | [6,1] | [6,2] | [6,3] | [6,4] | [6,5] | [6,6] | [6,7] |
| [7,0] | [7,1] | [7,2] | [7,3] | [7,4] | [7,5] | [7,6] | [7,7] |
| [8,0] | [8,1] | [8,2] | [8,3] | [8,4] | [8,5] | [8,6] | [8,7] |

### Activity 1.2.3

**Two-dimensional arrays (using nested lists)**

In Python, two-dimensional arrays are represented as nested lists (a list of lists) so the addresses are given as [row] [column] rather than [row,column]

* Copy and run this program. What happens and why?

numbRows = 4

numbCols = 6

myArray=[[0 for column in range(numbCols)] for row in range(numbRows)]

print(myArray)

* Change the value ‘0’ to ‘86’ and run the program again. What happens?
* Change the value back to ‘0’ and change numbRows to ‘9’ and numbCols to ‘5’. Run the program. What happens?
* Add these lines to the program as shown to assign values within the array. Run the program. What happens?

myArray[0][4] = 99

myArray[2][3] = 74

print(myArray)

* Add these rows to the program so that it prints out the array a row at a time. Explain how this works.

# print out a row at a time

for row in range(rowLength):

print(myArray[row])

### Activity 1.2.4

How to use two-dimensional arrays in Python (nested lists which start from zero)

|  |  |
| --- | --- |
| Task | Example |
| How to initialise a two-dimensional array |  |
| How to address an array element |  |
| How to assign values in a two-dimensional array |  |
| How to print a two-dimensional array |  |

### Activity 1.2.5

* Write a program that fills up a two-dimensional grid with the results of the multiplication table 10 x 10 and prints out the result.

### Activity 1.2.6

* Write a program to implement a ‘one player’ Battleships game. Use randint() to initialise the location of the ships. The player enters the index locations for their shots. The player scores one point for every direct hit on a ship. The player is allowed 10 ‘goes’ to hit as many ships as possible. The score is displayed at the end of the game.

# Week 2

## Lesson 1 Activities

### Activity 2.1.1

* Write an explanation of the client-server model below.
* Write an explanation of the peer-to-peer model below.

## Lesson 2 Activities

### Activity 2.2.1

Validation is the automatic checking of entered data by a computer program. Validation cannot check that the data entered is correct only that it is a reasonable value.

* Complete the first column of this table using the words listed below, to match the validation type with the correct description. Then give an example of each type in the third column.

presence range length type look-up

|  |  |  |
| --- | --- | --- |
| **Type of validation** | **Description** | **Example** |
|  | Checks the data entered is not too short or too long. |  |
|  | Checks that data has been entered. |  |
|  | Checks that the value entered falls within a given range. |  |
|  | Checks that the value entered is of the expected type. |  |
|  | Checks the entered value is a value that is expected. Checks value against a look-up list or string. |  |

### Activity 2.2.2

**Validation: length check**

* Write a program that asks the user to input a password and then uses a length check to make sure the password is at least eight characters long. If it is shorter than eight characters the user is asked to enter a different password.

*Hint: Use the len() function.*

### Activity 2.2.3

**Validation: presence check**

* Write a program that asks the user to enter a name and uses a presence check to make sure that an answer has been entered. If nothing has been entered the user is prompted again to enter a name.

**Extension:** The program should output an appropriate message when the user fails to enter any characters.

### Activity 2.2.4

**Validation: type check**

* Write a program that asks the user to enter their age and checks that they have entered an integer. It should display a message asking them to enter a whole number if they have not done so.

*Hint: Use the ‘try except else’ command.*

### Activity 2.2.5

**Validation: look-up check**

* Write a program that asks the user to enter an email address and then checks the string entered to make sure it contains an ‘@’. If it does not, the user is prompted again to enter an email address.

### Activity 2.2.6

**Validation: range check**

* Write a program that asks the user to enter a per cent (%) of charge left in their mobile phone. Use a range check to make sure the value is less 100 per cent (%) or more than 0 per cent (%). The program must ask the user to re-enter the value if it is outside the range.

**Extension:** Alter the program so it only allows integer numbers to be entered.

### Activity 2.2.7

**Try command: divide by zero error check**

* Write a program that asks the user for two numbers, then divides the numbers and displays the answer. If the program generates a divide by zero error, display a message to explain they entered a zero as the second number.

### Activity 2.2.8 (homework)

* Write a program that asks the user to enter their name, age and email address, using validation to ensure the data contains reasonable values. It should then display the data and ask the user if it is correct.

Use validation to make sure the user can reply ‘Y’, ‘y’ or ‘Yes’ and ‘N’, ‘n’, or ‘No’. The user should be allowed to re-enter the data if it is incorrect.

# Week 3

## Lesson 1 Activities

### Activity 3.1.1

* Make a note of what these acronyms mean.

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| **bps** |  |
| **Mbps** |  |
| **Gbps** |  |

### Activity 3.1.2

**Calculating the time required to transmit a file.**

* Calculate the times to download the following files on a 10 Mbps network connection:

**Remember:**

* Network data transmission speeds are measured in bits per second.
  + Transmission speeds are always in SI units, so Kbps is 1000 bits/sec.
* Data storage is measured in bytes:
  + 1 byte = 8 bits and 1 kibibyte = 1024 bytes.

To calculate the time to transmit a file you must convert the network speed and the file size into bits. Then use the formula:

time = size of file (in bits) / network speed (in bits)

|  |  |
| --- | --- |
| **File Size** | **Time to transfer** |
| **5 KiB** |  |
| **100 KiB** |  |
| **1 MiB** |  |

## Lesson 2 Activities

### Activity 3.2.1

**Default parameter values**

* Copy this program into a Python window.

def myFunction(name = “Fred”, surname = “Bloggs”, age = 21):

print("My name is {0} {1} and I am {2}".format(name, surname, age))

* Add these function calls to the program. Explain what happens when they are executed:

myFunction("George","Wales","0")

myFunction("Amelia","Jones")

myFunction("Vicky")

myFunction()

### Activity 3.2.2

**Subprograms in Python: writing a function**

* Copy this program into a Python window. Save the file as ‘testing\_functions.py’.

def function\_one(myName):

print("Hello",myName)

This is a subprogram that is expecting one parameter “myName”.

* Run this program from the IDLE shell by entering:

function\_one(“Edward”)

This will run function\_one and pass the argument ‘Edward’ into the subprogram.

* Run the function a few times passing different names into the file.

Remember that it is the name of the function that is used to run the function, not the name of the file in which you saved the function.

### Activity 3.2.3

* Write a function called name\_age in your testing\_functions.py file. Use parameters to pass a name and age into the function. The function should display this information appropriately on the screen. Run the function several times passing different names and ages.

### Activity 3.2.4

* Write a function called square in your testing\_functions.py file. Use a parameter to pass the length of a side into the function which must display the area of the square. Run the function several times passing different sizes of box.

### Activity 3.2.5

* Write a function called numbers in your testing\_functions.py file. Use parameters to pass two numbers into the function. The function must then count from the first number to the second number.

### Activity 3.2.6

* Copy and run this program:

# outputs a number

def display(number, answer, type):

print("This number {0} has been {2} the answer is {1}".format(number, answer, type))

# squares a number

def square(number):

answer = number \* number

return(answer)

# main program

amount = int(input("Please enter number: "))

for next in range(1, amount):

ans = square(next)

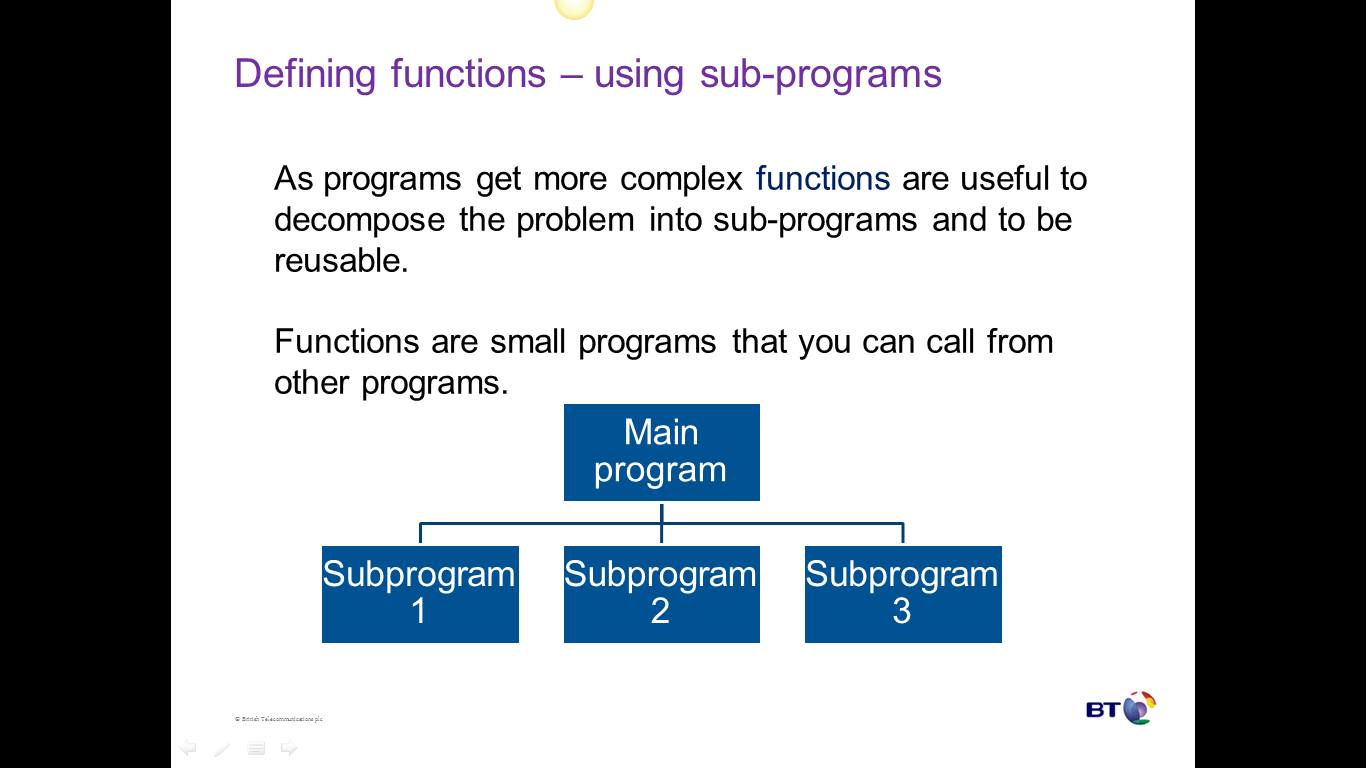
display(next, ans,"squared")

* Explain what this program does. Add appropriate comments to the program.
* Colour code the programming constructs, subprograms, return statement, parameters and arguments.
* Explain why the two functions are called in different ways.
* Amend the program to include another function that calculates the cube of the number and displays it.

### Activity 3.2.7

**Advantages of subprograms**

* What do you think are the advantages of using subprograms?



# Week 4

## Lesson 1 Activities

### Activity 4.1.1

**Network topologies**

Draw a labelled diagram of each of these network topologies:

* Bus
* Ring
* Star
* Mesh

Write notes on the advantages and disadvantages of each topology.

## Lesson 2 Activities

### Activity 4.2.1

**Local and global variables**

Variables can be ‘seen’ in different parts of a program. If a variable is defined inside a function then it usually cannot be seen outside that function and is called a local variable. If a variable is declared as being global then it can be seen everywhere. This is sometimes called the ‘scope’ of the variable.

* Copy and run this function:

def myFunction():

global variableOne

variableOne = 45

variableTwo = 2

* Run the commands at the prompt and explain the result.

>>>myFunction()

>>>print(variableOne)

45

>>>print(variableTwo)

### Activity 4.2.2

* Identify which are the local and global variables in this program. You may wish to copy and run the program.

def functionOne():

global answer

number1 = 90

number2 = 78

answer = number1 \* number2

def functionTwo():

global answer

number1 = 65

number2 = 3

answer = number1 / number2

The problem with using a global variable is that it can take different values in different functions. This makes it difficult to keep track of its correct value. Use global variables with care.

# Week 5

## Lesson 1 Activities

### Activity 5.1.1

* Research protocols. Write a simple definition of what a protocol is.
* What aspects of communication could a network protocol specify? An example is shown below.

|  |  |
| --- | --- |
| **Characteristic** | **Details** |
| **Error checking** | Each protocol could use a different type of error checking, for example parity. Error checking allows errors to be detected so that the corrupted data is rejected. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Lesson 2 Activities

### Activity 5.2.1

Design a program that lets a customer choose from a list of five different products and add this to their online basket. A discount is applied depending on the quantity ordered:

1-9, less 5%

10-29, less 10%

More than 30, less 15%

The final invoice should have 20% VAT added.

You should use constants.

Your teacher will split you into groups for this activity.

### Activity 5.2.2

Modify 5.2.1 to allow customers to choose more than one product. The program should loop until the customer has finished shopping.

# Week 6

## Lesson 1 Activities

### Activity 6.1.1

* Complete this table.

| **Protocol** | **What is it used for?** |
| --- | --- |
| **Ethernet** |  |
| **Wi-Fi** |  |
| **HTTP** |  |
| **HTTPS** |  |
| **FTP** |  |
| **POP3** |  |
| **SMTP** |  |
| **IMAP** |  |

## Lesson 2 Activities

### Activity 6.2.1

**The negative impact of computers on the environment.**

* Complete this table.

| **Area** | **Issue** | **Ways of limiting the environmental impact** |
| --- | --- | --- |
| **Manufacture** |  |  |
| **Usage** |  |  |
| **Disposal** |  |  |

### Activity 6.2.2

**Ways in which computers are helping to safeguard the environment**

* Research the ways in which computers help to safeguard the environment then fill in this table.

|  |  |
| --- | --- |
| **Computers are being used to…** | **For example** |
| Collect and analyse data about aspects of the environment, enabling us to get a better understanding of how the environment works, predict trends, manage scarce resources and plan ahead. |  |
| Protect wildlife. |  |
| Deal with forces of nature. |  |
| Develop low-impact alternatives to current activities that harm the environment. |  |

Research starting points:

* ‘Calculating the 'day after tomorrow'’, CS4FN – Short article on how computers are being used to predict the effect of climate change

<http://www.cs4fn.org/geography/climate.php>

* ‘Computing for the future of the planet’ article in ElectronicsWeekly.com

<http://www.electronicsweekly.com/news/business/information-technology/computing-for-the-future-of-the-planet-2009-07/>

* ‘Everything connected: the smart home in 2014’, article in the *Telegraph* 11 May 2014

<http://www.telegraph.co.uk/technology/internet/10542550/Everything-connected-the-smart-home-in-2014.html>

* ‘How to Classify a Million Galaxies in Three Weeks’, *Time*

<http://content.time.com/time/health/article/0,8599,1975296,00.html>

* ‘Tsunami Early Warning system’, CWarn.org

<http://cwarn.org>

* Google global impact awards – Digital Eyes and Ears for Wildlife Protection,

<https://impactchallenge.withgoogle.com/uk2013>

# Week 7

## Lesson 1 Activities

### Activity 7.1.1

* What is a protocol and why are protocols needed?

### Activity 7.1.2

* What is a protocol is and why are protocols needed? (Improved Class Answer)

### Activity 7.1.3

* What is a protocol suite?

### Activity 7.1.4

**TCP/IP**

* Complete the table.

|  |  |  |
| --- | --- | --- |
| **Layer** | **What does the layer do?** | **Protocols used at this layer** |
| **Application** |  |  |
| **Transport** |  |  |
| **Internet** |  |  |
| **Data link** |  |  |

### Activity 7.1.5

**What is a network packet?**

* Describe below what a packet is and what data a packet header can contain.
* You could include a diagram showing the structure of an IP packet to help with your explanation.

# Week 8

## Lesson 1 Activities

## Activity 8.1.1

**AND gate**

* Insert picture of the AND gate symbol here.

**Truth table for AND gate**

* Compete the outputs for the AND gate.

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **P** | **Q** | **R** |
| 0 | 0 |  |
| 0 | 1 |  |
| 1 | 0 |  |
| 1 | 1 |  |

* Explain what an AND gate does.

### Activity 8.1.2

**OR gate**

Insert picture of the OR gate symbol here

**Truth Table for OR gate**

Compete the outputs for the OR gate

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **P** | **Q** | **R** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |

Explain what an OR gate does.

## Lesson 2 Activities

### Activity 8.2.1

**Types of errors**

* Identify the error type from the description given. Choose from the words below. Then, give an example of each type of error from your own experience.

logic syntax runtime

|  |  |  |
| --- | --- | --- |
| **Type of error** | **Description** | **Example from your programming experience** |
|  | Errors detected during the program execution, often from mistakes in the algorithm used or type of data used. |  |
|  | Errors that occur when the program statements cannot be understood because they do not follow the rules of the programming language. |  |
|  | Errors in the design of the program, such as the use of the wrong programming control structures or the wrong logic in condition statements. This type of error may not produce error messages just the wrong answer. |  |

### Activity 8.2.2

**Trace tables**

* Complete the trace table for the following program:

for next in range(1,10,2):

number1 = next

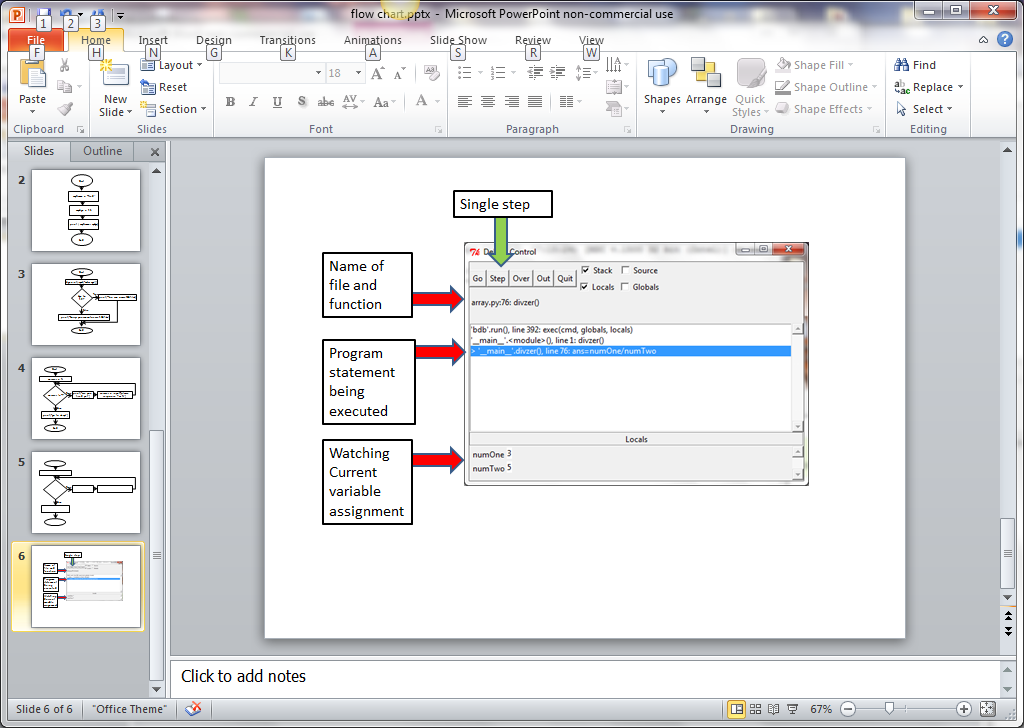
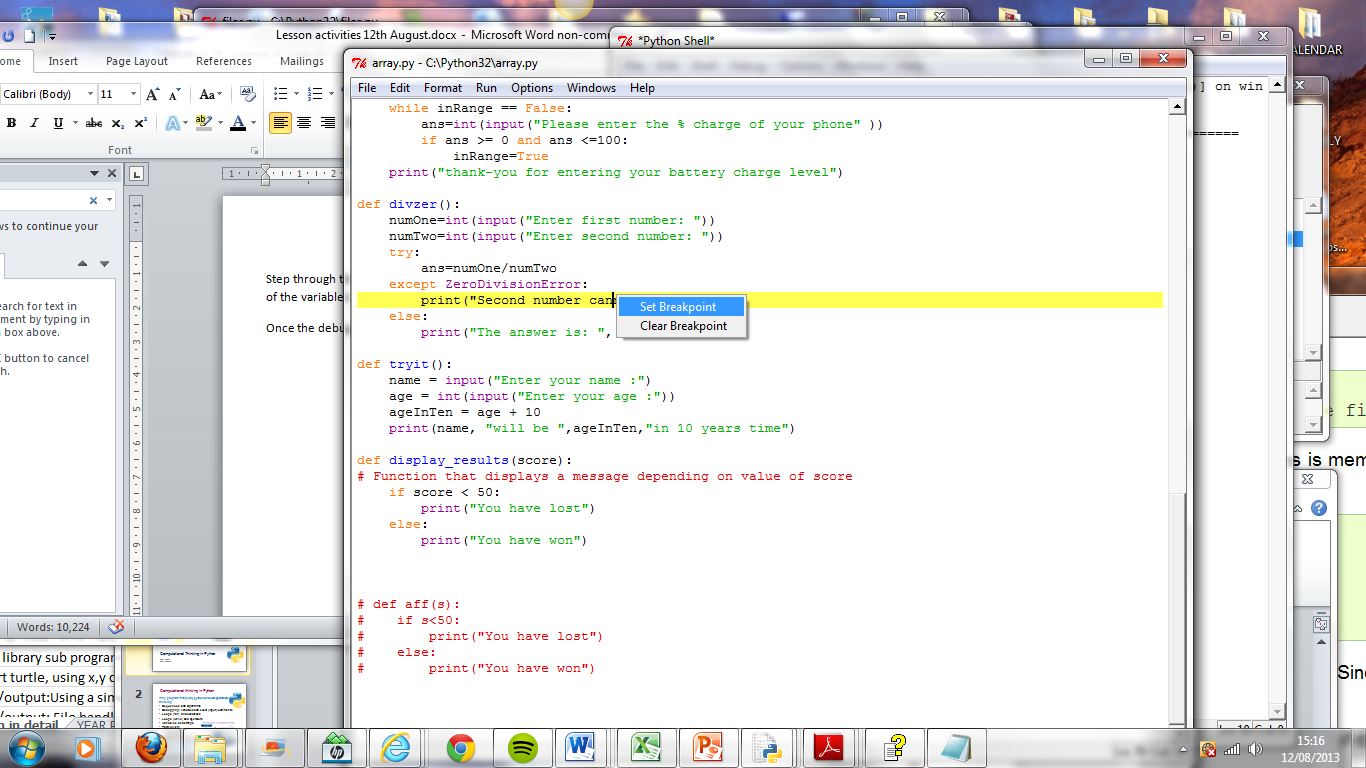
number2 = next \* next

number3 = next / 2

|  |  |  |  |
| --- | --- | --- | --- |
| **next** | **number1** | **number2** | **number3** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Activity 8.2.3

**Exploring the IDLE debugger**

* Open Python IDLE with an existing program file.
* Start the IDLE debugger using Debug/Debugger. The IDLE shell will display [DEBUG ON] to show you are in debug mode. To turn off this mode select Debug/Debugger.
* Start running your program. The Debug Control window will be displayed:
* Step through the program by clicking on ‘Step’. The current command being executed and the current values of the variables will be displayed. Predict what you expect these to be.
* Once the debugger has been switched on break points can be set by right clicking in the program and selecting ‘Set Breakpoint’.
* When you run the program it will stop at the break point and allow you to view the variable values at this point. You can set more than one break point in a program.
* Use the Python IDLE debugger to explore some of your existing programs.
* Use the Python IDLE debugger to help you debug errors in your programs.

# Week 9

## Lesson 1 Activities

### Activity 9.1.1

**NOT gate**

* Insert picture of the NOT gate symbol here:

**Truth Table for NOT gate**

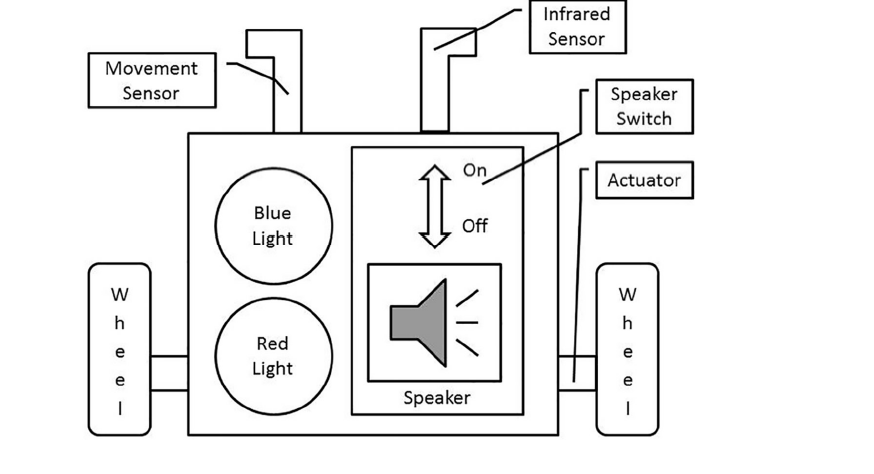
* Compete the output for a NOT statement.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0 |  |
| 1 |  |

* Explain what the NOT operator does:

### Activity 9.1.2

* Study this block diagram of a robot.



* Assume R is the red light, B is the blue light and S is the speaker switch.
* Construct a truth table for a NOT operator that is used to reverse the state of the blue light.

|  |  |
| --- | --- |
| **R** | **NOT(R)** |
|  |  |
|  |  |

* Complete this table for the logic statement S =(NOT R) AND B.

|  |  |  |  |
| --- | --- | --- | --- |
| **R** | **B** | **NOT R** | **Q** |
| 0 | 0 |  |  |
| 0 | 1 |  |  |
| 1 | 0 |  |  |
| 1 | 1 |  |  |

* Explain what this logic statement does.
* Construct a logic statement to determine if either light is on at the same time as the speaker (S) is on.

## Lesson 2

### Activity 9.2.1

Web browser module in Python 3

This module allows Python to open web pages from programs. More details can be found at <http://docs.python.org/2/library/webbrowser.html>.

webbrowser.open(URL) opens a web page given a URL as a string in your default web browser.

>>>webbrowser.open("http://www.bbc.co.uk")

* Copy and run this program:

def webpage():

import webbrowser

web1 ="http://www.bbc.co.uk/"

web2="http://en.wikipedia.org/wiki/Main\_Page"

web3= "http://www.nasa.gov/"

answer = int(input("""

1 To open BBC

2 To open Wikipedia

3 To open NASA

"""))

if answer ==1:

webbrowser.open(web1)

elif answer == 2:

webbrowser.open(web2)

else:

webbrowser.open(web3)

#main program

webpage()

* Amend the program to open other web pages.

### Activity 9.2.2

winsound module in Python 3

This module allows Python to play sounds on Windows computers. Try out the commands below to explore the winsound module.

**winsound.Beep(frequency,duration)** makes a sound at the frequency given, for the amount of time given in duration. Frequency is in hertz from 32 to 32,767. Duration is in milliseconds.

>>>winsound.Beep(1000,500)

* Experiment with different values of frequency and duration.
* Copy and run this program. Explain what it does.

def sound():

import winsound

frequency = 100

duration = 200

number = 40

for index in range (40):

winsound.Beep(frequency,200)

frequency = frequency + 50

#main program

winsound()

* Experiment entering different frequency and duration values.

**winsound.PlaySound(filename, flags)** plays the sound file specified by the filename. This can be used to play the Windows sound files (for Windows operating systems only), which have been set up by the operating system.

>>>import winsound

>>>winsound.PlaySound("SystemExit", winsound.SND\_ALIAS)

This plays the ‘SystemExit’ sound. Other sounds are available – experiment to find out what sounds these filenames give.

|  |
| --- |
| “SystemAsterisk” |
| “SystemExclamation” |
| “SystemExit” |
| “SystemHand” |
| “SystemQuestion” |

* Write a program that plays a different note depending on a number in the range 1 to 10, which is entered by the user.

### Activity 9.2.3

Math module in Python 3

* Try out the commands below to explore the math module. A full explanation is given here: <http://docs.python.org/2/library/math.html>
* Import the math module and list the directory of mathematical functions:

>>>import math

>>>dir(math)

['\_\_doc\_\_', '\_\_name\_\_', '\_\_package\_\_', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']

* Use mathematical constants such as pi.

>>>math.pi

3.141592653589793

* Raise a number to a power. The first number is the number and the second is the power.

>>>ans=math.pow(2,3)

>>>ans

8.0

* Find the square root of a number.

>>>ans=math.sqrt(1024)

>>>ans

32.0

Challenges using the math module:

* Find the area of circle with a radius of 19.7.
* Find the square root of 1048576.
* Find the value of 5 to the power 4.
* A right-angled triangle has sides of 5 and 6. Find the length of the other side.
* Find examples of other math functions that you may find useful.

# Week 10

## Lesson 1 Activities

### Activity 10.1.1

* What gates do these Truth Tables represent?

**Truth Table A**

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| P | Q | R |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Truth Table B**

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| P | Q | R |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Truth Table C**

|  |  |
| --- | --- |
| **Inputs** | **Output** |
| Q | R |
| 0 | 1 |
| 1 | 0 |

### Activity 10.1.2

* Construct a logic statement to represent the system detailed below:

A greenhouse heating system will only operate if:

* The on/off switch is set to on (S)
* The door is closed – this is monitored by a sensor and will output TRUE when the door is closed (D)
* The windows are closed – again monitored by a sensor that will output TRUE when the windows are closed (W)

Your logic statement should use the symbols S, D and W

### Activity 10.1.3

Boolean expressions are used to represent logical behaviours.

The security lights in a car park is controlled by an embedded system.

* There is a power (P) switch to control the lights
* A light (L) sensor will turn the car park lights on when it gets dark, provided the power switch is in the on position
* An override (O) switch will turn the lights on, regardless of the light sensor, provided the power switch is in the on position
* Write a logic statement to represent the above system’s behaviour. Use the symbols P, L and O.

## Lesson 2 Activities

### Activity 10.2.1

Read Stephen Waddington’s blog post: ‘My mobile phone knows more about me than my family’ and answer these questions.

http://wadds.co.uk/2014/02/02/mobile-phone-family/

|  |  |
| --- | --- |
| How might advertisers make use of location information? |  |
| What are the benefits of making location information available to others? |  |
| What are the drawbacks? |  |
| What does the algorithm developed by the research team from Birmingham University do? |  |

### Activity 10.2.2

Read James Ball’s article: ‘Me and my data: how much do the internet giants really know?’ and answer these questions. <https://www.theguardian.com/technology/2012/apr/22/me-and-my-data-internet-giants>

|  |  |
| --- | --- |
| What personal information does Google hold about James? |  |
| What other information is there about James that he can’t find out about? |  |
| Google uses cookies to determine which adverts to display. What is a cookie? |  |

# Week 11

## Lesson 1 Activities

### Activity 11.1.1

How could text be stored within a computer system?

* Discuss the following questions and make notes about the conclusions you come to.

1. How would storing lowercase/capitals work?
2. How would storage of numbers work?
3. Other than letters and numbers, what else would need to be stored?

### Activity 11.1.2

* Discuss:

If every computer system used a different system for storing text what problems would this cause?

For example, Computer 1 would store ‘a’ as 97 (decimal) and ‘b’ as 98 (decimal) but Computer 2 stored ‘a’ as 120 (decimal) and ‘b’ as 121 (decimal).

Note any problems you can think of below:

### Activity 11.1.3

* Use the World Wide Web to find out the answers to the following questions:

1. What does ASCII stand for?
2. What is ASCII?
3. Complete this table to show the ASCII code for the given characters.

| **Character** | **Binary** | **Decimal** |
| --- | --- | --- |
| **a** |  |  |
| **b** |  |  |
| **c** |  |  |
| **A** |  |  |
| **B** |  |  |
| **C** |  |  |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |

1. Explain how ASCII is organised
2. Other than text and numbers what else is stored? Why?

## Lesson 2 Activities

Recall James Ball’s article: ‘Me and my data: how much do the internet giants really know?’ and answer these questions.

### Activity 11.2.1

**Security threats to data.**

| **Security threat** | **Description, including threat to personal data** | **What organisations can do to prevent them** |
| --- | --- | --- |
| **Viruses** |  |  |
| **Hackers** |  |  |
| **Phishing** |  |  |
| **Trojan horses** |  |  |
| **Worms** |  |  |
| **Impersonation** |  |  |
| **Password cracking** |  |  |
| **Denial of service attacks** |  |  |
| **Eavesdropping** |  |  |
| **Network spoofing** |  |  |
| **Mail bombing** |  |  |
| **Macro viruses** |  |  |

# Week 12

## Lesson 2 Activities

## The evolution of mobile communications from 1G to 4G

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1G** | **2G** | **3G** | **4G** |
| Introduced in |  |  |  |  |
| Speed |  |  |  |  |
| Bandwidth |  |  |  |  |
| Capacity |  |  |  |  |
| Services |  |  |  |  |