**Lesson 14 activities**

**Activity 14.1**

**Pseudocode**

Pseudocode, or ‘mock’ code, is another way of describing a program. You will need a copy of the pseudocode used in this course (Appendix B of the specification) to answer these questions.

* What does this pseudocode do?

RECEIVE myName FROM (STRING) KEYBOARD

RECEIVE myAge FROM (INTEGER) KEYBOARD

SET AgeInTen TOmyAge + 10

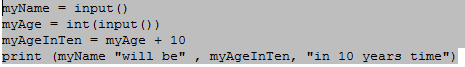
SEND myName “will be” AgeInTen “in 10 years’ time” TO DISPLAY

Input your name & age

Add 10 years to your age

In 10 years, I’ll be 10+ to your age.

* Write the Python code for this program.



**Activity 14.2**

* What does this pseudocode do?

SET score TO 119

IF score < 50 THEN SEND “You have lost” TO DISPLAY

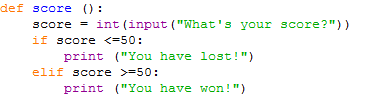
ELSE SEND “You have won” TO DISPLAY

END IF

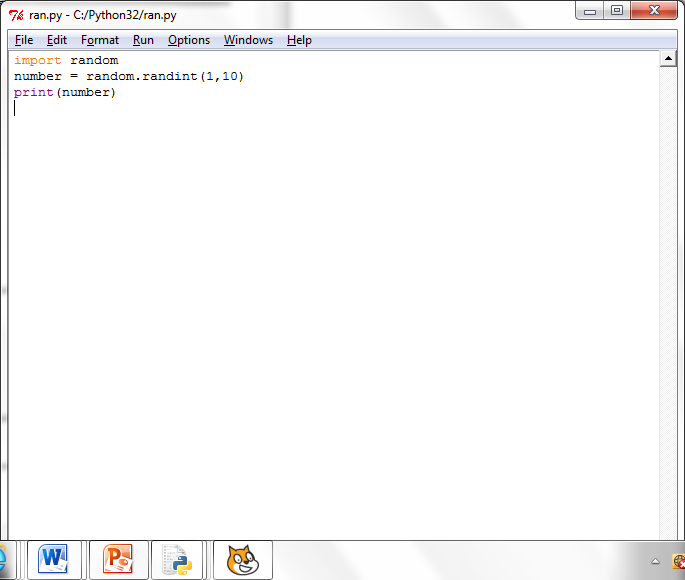
If your score is below 50 then “You have Lost”

Otheriwse, “You have won”

* Write the Python code for this program.

**Activity 14.3**

**random() function**

* Copy and run this program code:
* Run the program a few times. What does it do?

It gives me a different number each time.

* What is meant by ‘import random’?

It creates a random number.

* What happens if you alter the values to 1, 100?

The range of the numbers increases.

**Activity 14.4**

Write a program that acts like a dice. After each ‘throw’ of the dice it should ask if the player wishes to continue and stop when they enter ‘Y’.

**Activity 14.5**

Write a program that gives the user 10 chances to guess a number between 1 and 10. It then compares their answer with a randomly generated number and stops the program when the number generated matches the number guessed or when the user is out of guesses.

**Lesson 15 activities**

**Activity 15.1**

**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?
* How would you display “China”?
* How would you display “Japan”
* How would you address the third item in the list?
* Copy and run this program. Explain how it works.

for name in countries:

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

* 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 15.2**

**Battleships: a game using two-dimensional array addressing (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 have five ships:

* A battle ship that takes up five index spaces
* A cruiser that takes up four index spaces
* A submarine that takes up three index spaces
* A destroyer that takes up two index space
* 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 play 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 15.3**

**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?

rowLength=4

columnLength=6

myArray=[[0 for row in range(rowLength)] for column in range(columnLength)]

print(myArray)

* Change the value ‘0’ to ‘86’ and run the program again. What happens and why?
* Change the value back to 0 and change rowLength to 9 and columnLength 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 and why?

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 it works.

# print out a row at a time

for row in range(rowLength):

print(myArray[row])

**Activity 15.4**

|  |  |
| --- | --- |
| **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 15.5**

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 15.6**

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.