# Python Coding

The programming assignment gives the students the opportunity to demonstrate the coding skills that they have acquired during the course. This selection of programs should introduce the skills to the students, enabling them to tackle the programming assignment.

**1 Life Decisions (Magic 8-ball)**

Skills: Import module, Python Lists, Randomly choose from a List, one line comments

#Life decisions

import random

decisions=["Stay in bed","Go for a walk","Go clothes shopping","Hang out with friends","Do some programming!"]

print("What to do today: ",random.choice(decisions))

The program above is identical to the one below but you can space things out to make it easier to read. Do not indent because this has a special meaning in Python.

#Life decisions

import random

decisions=[

"Stay in bed",

"Go for a walk",

"Go clothes shopping",

"Hang out with friends",

"Do some programming!"

]

print("What to do today: ",random.choice(decisions))

**2a One-Roll Dice**

New Skills: Randomly choose a number in a given range

Notice that to get a range of 1 to 6 inclusive the function needs 1,7 for its range.

# One-Roll Dice

import random

print("Roll a six to win")

number=random.randrange(1,7)

print("Roll is: ", number)

**2b One-Roll Dice (Am I a Winner?)**

New Skills: If….Then

Notice the indentation. Only if the number *IS* 6 will the indented line be executed.

# One-Roll Dice

import random

print("Roll a six to win.")

number=random.randrange(1,7)

print("Roll is: ", number)

if number==6:

print("Winner!!!!")

**2c One-Roll Dice (Choose which number wins)**

New Skills: input, Integer variable

# One-Roll Dice

import random

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

print("Roll is: ", number)

if number==users\_guess:

print("You win!!!!")

**2d One-Roll Dice (What if I’m not a Winner?)**

New Skills: If….Then….else

# One-Roll Dice

import random

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

print("Roll is: ", number)

if number==users\_guess:

print("You win!!!!")

else:

print("You lose")

Notice the indentation. If it’s true do one thing, otherwise (else) do something else.

**3a Guess the Roll (Don’t want to keep pressing Run?)**

New Skills: while, Boolean true

# Guess the Dice

import random

while True:

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

print("Roll is: ", number)

if number==users\_guess:

print("You win!!!!")

else:

print("You lose")

Notice the indentation. while True will repeat everything that is indented forever! You can stop the program by selecting Cancel.

**3b Guess the Roll (Count the number of rolls)**

New Skills: increment a variable. Format printout

# Guess the Dice

import random

spin\_count=0

while True:

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count, number))

if number==users\_guess:

print("You win!!!!")

else:

print("You lose")

{} is a place holder for the data in the variable list at the end. The order of the data is important.

**3c Guess the Roll (Count the number of WINS!)**

New Skills: Alternative way of incrementing a variable (+=)

# Guess the Dice

import random

spin\_count=0

wins=0

while True:

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count,number))

if number==users\_guess:

wins += 1

print("This is win number {}".format(wins))

else:

print("You lose")

**3d Guess the Roll (Percentage wins)**

New Skills: Simple calculation (division)

# Guess the Dice

import random

spin\_count=0

wins=0

while True:

users\_guess=int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count,number))

if number==users\_guess:

wins += 1

win\_percent=wins/spin\_count

print("This is win number {}. You win percentage is {}".format(wins,win\_percent))

else:

print("You lose")

**3e Guess the Roll (Percentage to 2 DP)**

New Skills: format to 2 decimal places, floating point variable

# Guess the Dice

import random

spin\_count=0

wins=0

while True:

users\_guess= int(input("Guess the roll of the dice"))

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count,number))

if number==users\_guess:

wins += 1

win\_percent=wins/spin\_count

print("This is win number {}. You win percentage is {:.2f}".format(wins,win\_percent))

else:

print("You lose")

{:.2f} prints the floating point number to 2 DP.

**3f Guess the Roll (I want to stop)**

New Skills: while (expression is true) repeat

# Guess the Dice

import random

spin\_count=0

wins=0

users\_guess=5

while users\_guess>0:

users\_guess= int(input("Guess the roll of the dice (enter 0 to stop)"))

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count,number))

if number==users\_guess:

wins += 1

win\_percent=wins/spin\_count

print("This is win number {}. You win percentage is {:.2f}".format(wins,win\_percent))

else:

print("You lose")

Whilst ‘users\_guess’ remains greater than zero the indented code will be repeated.

**You should be able to tackle the Bingo Game exercise at the end of this document.**

**4 Validation (Only accept integers between 1 and 6, inclusive)**

New Skills: Validation, try, raise, except, ValueError

I would guess that earlier the program would have crashed when the user entered nothing instead of a number between 1 and 6. Validation would trap that error and stop the program from crashing.

# Interlude - Validation

users\_guess=0

attempts=0

prompt="Enter a number between 1 and 6 inclusive"

while not(1 <= users\_guess <=6):

try:

attempts +=1

users\_guess= int(input(prompt))

if not (1 <= users\_guess <= 6):

raise ValueError()

except ValueError:

prompt="Invalid entry. It must be 1 to 6"

print("You took {} attempts to enter a correct number".format(attempts))

**5a Guess the Roll (Only accept integers between 1 and 6, inclusive)**

New Skills: Self-defined functions, local variables, multi-line comments (‘’’)

# Guess the Dice

import random

#----------------------------------------

def ask\_user(prompt):

ug=0

while not(1 <= ug <=6):

try:

ug= int(input(prompt))

if not (1 <= ug <= 6):

raise ValueError()

except ValueError:

prompt="Invalid entry. It must be 1 to 6"

return ug

#----------------------------------------

'''

Discussion: local variables.

prompt and ug are local to the function and their values are

not 'seen' outside of it

but values can be passed in and out of it.

ie. A message is passed in and the user's guess is out of it.

The string "Guess the roll of the dice" is passed into

the function and becomes prompt,

whilst ug is returned from the function and passes

the value to user\_guess

'''

#----------------------------------------

spin\_count=0

wins=0

while True:

users\_guess= ask\_user("Guess the roll of the dice") # see discussion on local variables above

number=random.randrange(1,7)

spin\_count = spin\_count +1

print("Number of spins {}. The roll is: {} ".format(spin\_count,number))

if number==users\_guess:

wins += 1

win\_percent=wins/spin\_count

print("This is win number {}. You win percentage is {:.2f}".format(wins,win\_percent))

else:

print("You lose")

Local variables.

prompt and ug are local to the function and their values are not 'seen' outside of it but values can be passed in and out of it.

ie. A message is passed in and the user's guess is out of it.

The string "Guess the roll of the dice" is passed into

the function and becomes prompt,

whilst ug is returned from the function and passes

the value to user\_guess.

**6a Read My Timetable (The file must already exist)**

New Skills: open, readlines, close

In Notepad create a file of subjects and subject teachers for each lesson of the day. Save the file as a text file, with the day of the week as its name. Save it in the same directory as the Python program.

Eg. Here is Tuesday.txt

Maths, BA

English, JN

French, MD

PE, TF

Science, RF

Woodwork, JP

#Read My Timetable

#----------------------------------------

def readfile(day):

filename=day+".txt"

channel = open(filename,"r+")

lesson=channel.readlines()

channel.close()

return lesson

#----------------------------------------

day=input("Day of week")

lesson=readfile(day)

print(lesson)

The function readfile() has the day of the week passed into it.

.txt is added to the day of the week and a channel is opened onto that file. Each line of the file is read into the List called lesson.

Call the function readfile() and pass the variable day into it. The result of the function (ie. The List of lessons) will be put into the LIST called lessons.

**6b Read My Timetable (The file must already exist)**

New Skills: for … in list

#Read My Timetable

def readfile(day):

filename=day+".txt"

channel = open(filename,"r+")

lesson=channel.readlines()

channel.close()

return lesson

#----------------------------------------

day=input("Day of week")

lesson=readfile(day)

# print(lesson)

print("Day - ",day,"\n")

for i in lesson:

print(i)

The List called ‘lesson’ is read in one item at a time and becomes the variable i. This could be any variable name.

**6c Read My Timetable (The file must already exist)**

New Skills: for … in list, split()

#Read My Timetable

def readfile(day):

filename=day+".txt"

channel = open(filename,"r+")

lesson=channel.readlines()

channel.close()

return lesson

#----------------------------------------

day=input("Day of week")

lesson=readfile(day)

print("Day - ",day,"\n")

for i in lesson:

(subject, teacher)=(i.split(","))

print("Subject {}, Teacher {}".format(subject, teacher))

Split(), splits each list item in two, around the comma (in brackets). The left half goes into the variable subject and the right into teacher.

**6d Read My Timetable (The file must already exist)**

New Skills: None!

#Read My Timetable

def readfile(day):

filename=day+".txt"

channel = open(filename,"r+")

lesson=channel.readlines()

channel.close()

return lesson

#----------------------------------------

day=input("Day of week")

lesson=readfile(day)

print("Day - ",day,"\n" )

lesson\_num=1

for i in lesson:

(subject, teacher)=(i.split(","))

print("Lesson {} - {} {}".format(lesson\_num,subject, teacher))

lesson\_num +=1

Add the lesson number to the output.

Exercises

1. Develop a ‘Bingo Game’ (See example below). Do NOT go straight into code. For each task write out your plan using bullet points, or any method that you know to describe processes. Your final code may look vastly different from your original plan.  
   * Task 1 – Picture Bingo
     + The bingo game only has 20 object names
     + The bingo cards have 5 pictures of objects
     + An object name is picked at random, called out but is NOT removed from the list. The player then crosses that object off their picture card
     + When a player has all of the 5 pictures crossed off, they must shout house and the game ends
   * Task 2 – Improved Picture Bingo
     + As Task 1 but while ‘House’ has not been called, loop the picking and calling process
   * Task 3 – Non-repeating Objects
     + As Task 2 but when an object is picked, it IS REMOVED from the list. Use Google to find out how to remove an item from a List.
   * Task 4 – Normal Bingo
     + Replace the 20 object names with 90 number names
     + Replace the picture-bingo cards with traditional 15 number bingo cards
   * Task 5 – Check winner
     + When a player calls House, their ticket has to be checked against a list of the numbers called. How could you do this?
   * Task 6 – Jackpot
     + If a house is called within a certain number of picks (eg 55 numbers or less), the winner will also receive a Jackpot prize.
2. Adapt the solution for Task 3 from exercise 1 to read in a list of 20 objects from a text file.  
   * Task 1 – Read object names from files
     + Create a text file of 20 objects names. Save it with an appropriate name
     + Adapt the solution for Task 3 from exercise 1 to read in a list of 20 objects from a file

Picture Bingo

## Method 1

* Cut out the definitions to produce tiles
* Give out the Picture Card and the definition tiles
* When an object is called out, cover the picture with the definition

## Method 2

* Only give out the definitions as a bingo card
* When an object is called out, put a line through the definition

This may help: <https://www.computerhope.com/jargon.htm>

# Definitions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Volatile **memory** that the computer can access faster than CD, HD or SSID | Supplies components in a computer with **power** | An output device that **displays** video images and text | Handheld hardware **input device**that controls a cursor in a GUI and can move and select text, icons, files, and folders. | A coin cell **battery** used to power the memory that stores the system time and date and also the system hardware settings for the computer. |

# Picture Bingo Cards

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\andy\AppData\Local\Microsoft\Windows\INetCache\IE\3YFCHLD1\memoria-ram[1].jpg | C:\Users\andy\AppData\Local\Microsoft\Windows\INetCache\IE\3YFCHLD1\220px-PSU-Open1[1].jpg |  | C:\Users\andy\AppData\Local\Microsoft\Windows\INetCache\IE\ZQUWPRP8\1050.Dell%20S2340T%20multi-touch%20Windows%208%20monitor%20(front).jpg-550x0[1].jpg |  |
|  |  | C:\Users\andy\AppData\Local\Microsoft\Windows\INetCache\IE\YZBRYQYH\im_mouse[1].png |  | C:\Users\andy\AppData\Local\Microsoft\Windows\INetCache\IE\A0NOS3UI\GLtCO[1].jpg |