Computing Y11 – Python

The de Ferrers Trust

Python -> English	
print("hello!")	Prints a value on screen (in this case, hello!)
input("")	Inputs a value into the computer.
x = input("")	Inputs a value and stores it into the variable x.
x = int(input(""))	Inputs a value into x, whilst also making it into an integer.
answer = x + y	Saves the result of x and y added together in a variable named answer.
print(str(x))	Prints the variable x, but converts it into a string first.
print("Hello", "World")	Prints the two strings concatenated with a space between. This code would output "Hello World".
age = 12 print("Age: " + str(age))	The + joins together two variables when printing. Str has to be used to cast age to be a string. This code will output "Age: 12".
if name == "Fred":	Decides whether the variable 'name' ha a value which is equal to 'Fred'.
else:	The other option if the conditions for an if statement are not met (eg. name = 'Bob' when it should be Fred)
elif name == "Tim":	elif (short for else if) is for when the first if condition is not met, but you want to specify another option.
# COMMENT	# is used to make comments in code – any line which starts with a # will be ignored when the program runs. They are used to describe the code to a programmer.
for i in range(0,10): # WRITE CODE HERE	Repeats any code indented after this line a set number of times, in this case, 10.
while x < 10: # WRITE CODE HERE	Repeats any code indented after this line until a condition is met, in this case x becoming equal to or greater than 10.
list = ["","",""]	Creates a variable and makes it an array – a list which can store many values.

Data types

Data Type	This indicates how the data will be stored. The most common data types are integer, string, and float/real.	Casting code
String	A combination of letters, numbers or characters. (eg, Hello, WR10 1XA)	str(x)
Integer	A whole number. (eg. 1, 189)	int(x)
Float/ Real	A decimal number, not a whole number. (eg. 3.14, -26.9)	float(x)
Boolean	1 of 2 values. (eg. True, False, Yes, No)	bool(x)
Char	A single character	char(x)

Comparative operators

==	Equal to
!=	Not equal to (or different to)
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Arithmetic operators

Operation	Symbol	Example	Output
Addition	+	2 + 10	12
Subtraction	-	9 – 6	3
Multiplication	*	5 * 4	20
Division	/	5 / 2	2.5
Floor Division	//	7 // 2	3
Remainder	%	7 % 3	1

Key vocab	
Python	A programming language used to write programs.
Shell	The place where code is run.
Code editor	The place where code is written.
Programming	The process of writing computer programs.
Algorithm	A set of rules/instructions to be followed by a computer system.
Flowchart	A visual method of planning an algorithm using symbols.
Pseudocode	A language similar to English which is used to plan algorithms.
Code	The instructions that a program uses.
Sequence	Parts of the code that run in order and the pathway of the program reads and runs very line in order.
Selection	Selects a pathways through the code based on whether a condition is true.
Iteration	Code is repeated (looped), either *while* something is true or *for* a number of times.
Variable	A value that will change whilst the program is executed. (eg. temperature, speed)
Function	A collection of code that works outside the main program. These are created to speed up programming. They can be called from a single line of code at any time.
Comparative Operator	A symbol used to compare multiple values.
Arithmetic operator	A symbol used to manipulate numerical values.
Syntax	The punctuation/way that code has to be written so that the computer can understand it. Each programming language has its own syntax.
Syntax error	An error produced when the computer cannot understand the code which has been written.
Logic error	An error produced when a program is understood by the computer but does not perform as theprogrammer expects.

Finding errors – follow these steps

1. Have you checked that you have closed all brackets correctly?
2. Have you checked that you have closed all quotes correctly?
3. Are your variable names spelt in the same way consistently? Remember that Python is case sensitive
4. Have you remembered to use commas to separate the variables inside print?
5. Have you used quotes around strings which you want to print out word for word?
6. Have you used int or float on number inputs?

Addition example code

```
number1 = int(input("Input the first number :"))
number2 = int(input("Input the second number :"))
answer = number1 + number2
print("The answer is " + str(answer))
```

The code above takes two number inputs and stores them as variables called number1 and number2. It then adds these together and saves them in a variable called answer. The final line prints the answer out in a sentence.

Selection example code

```
fav_num = int(input("Pick a number between 1 & 10..."))

if(fav_num == 7):
   print("Good guess!")
elif(fav_num < 7):
   print("Too low!")
else:
   print("Too high!")
```

The code above inputs a number. If the number is 7 it will print “Good guess!”, if it is less than 7 it will print “Too low!” and for anything else it will print “Too high!”.