Year 7

Computing

5. Programming with Python

STUDENT	
TEACHER	
CLASS	

WORKING AT GRADE	
TERM TARGET	
YEAR TARGET	

GRADE FOR THIS TOPIC	

The long answer questions in this booklet are designed to stretch and challenge you. It is important that you understand how they should be answered. You should structure your answer like this:

1st Paragraph – should explain the key term e.g. give a definition.

2nd Paragraph – should make a point (could be an advantage or disadvantage) and explain the point fully giving an example where necessary.

3rd Paragraph – should make another point (could be an advantage or disadvantage) and explain the point fully giving an example where necessary.

4th Paragraph – should make a point (could be an advantage or disadvantage) and explain the point fully giving an example where necessary.

You should have at least 1 advantage and 1 disadvantage.

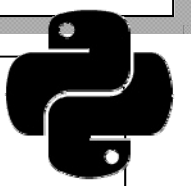
Progress against termly target												
ABOVE												
ON												
BELOW												
TERM	**1**		**2**		**3**		**4**		**5**		**6**	

Learning Outcomes			
	Levels		
Lesson	**3**	**4**	**5**
1 **Introduction to Python**	I can create programs that implement algorithms to achieve given goals.	I can design, write and debug modular programs using procedures.	I have practical experience of a high-level textual language,
2 **Variables and user inputs**	I can declare and assign variables.	I can use a variable and relational operators within a loop to govern termination.	I have practical experience of a high-level textual language, including using standard libraries when programming.
3 **Selection statements**	I can use a sequence of selection statements in programs, including an if, then and else statement.	I know the difference between, and appropriately I can use if and if, then and else statements.	I can use a range of operators and expressions e.g. Boolean, and applies them in the context of program control.
4 **Until loop**	I can use post-tested loops e.g. 'until',	I can use a variable and relational operators within a loop to govern termination.	I can select the appropriate data types.
5 **Arithmetic operators**	I can create programs that implement algorithms to achieve given goals.	I can design, write and debug modular programs using procedures.	I can use a range of operators and expressions e.g. Boolean, and applies them in the context of program control.
6 **Procedures**	I can create programs that implement algorithms to achieve given goals.	I know that a procedure can be used to hide the detail with sub-solution (procedural abstraction).	I know that programming bridges the gap between algorithmic solutions and computers.

1. Introduction to Python

What do you think programming is?

Can you name any programming languages?



```
print("Hello World")
```

What colours are used when you type the above code into IDLE?

print =

( ) =

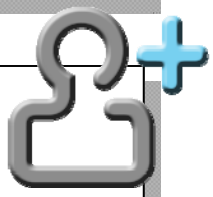
"Hello World" =

Run the program by pressing F5 - you will be prompted to save.

Challenge

Can you change the above code so the program prints your name?

Write your code below.



Extension

Try adding the following sentence in a print statement:

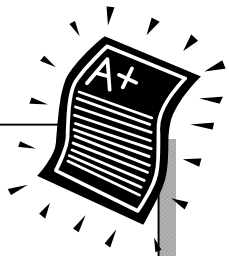
The quick brown fox jumps over the lazy dog.

At the end of the word fox before the closing speech mark add the following:

\n

What happened? What does the \n do?

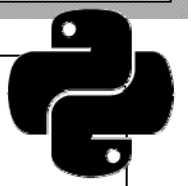
Self Assessment: R A G	Exit Ticket: What does the Print statement do?



2. Variables and user inputs

Correct this code so it works correctly:

```
print(“Algebraic!”)
Print(Mathmatical!”)
Print(“Oh yeah!”
```



```
input(“What is your name?”)
```

What happens when you run the above code in IDLE?

Challenge

Adapt the code to the following:

```
name = input("What is your name?")
print(name)
```

What happens this time?

Why do you think this is?

Extension

Adapt the second line of code to read:

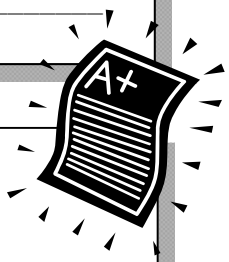
```
print(name * 2)
```

What happens?

Why do you think this happens?

Can you work out how to add "Your name is" to the string out putted? Try putting a + or a , between the extra words and the variable name. Write your code below when it works.

Self Assessment: **R A G**	Exit Ticket: What does the input function do?



3. Selection statements

Correct this code so it works correctly:

```
name = inpt("What's your name?")
print "Hello", name)
```

```
pet = input("What pet do you have?")
if pet == "Cat":
    print("You have a cat!")
```

Run the code. What happens if you type in Cat?

Challenge

Run the code again. This time type in Dog, what happens?

Why do you think this is?

```
name = input("What is your name?")
if name == "Dave":
    print("Hello Dave")
elif name == "Peter":
    print("Hello Peter")
else:
    print("Hello")
```

Challenge

Run the code. Try typing in Dave, then Peter and then another name. In the test table below explain what you would expect to happen and what actually happened for each run.

Test Number	Input	Expected Output	Actual Output
1	Dave		
2	peter		
3			

What happened when you typed your own choice of name? WHY?

Add the following code to the second line of your program:

```
name = name.capitalize()
```

What does this do?

Extension

Adapt the code to request the users age (don't forget to type cast for int!).

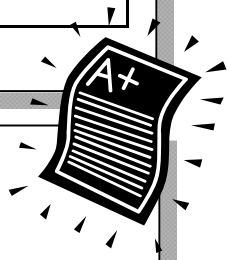
```
age = int(input("How old are you?")
if age > 18:
    print("You are old enough!")
```

Try running your code - use a test table to find out what is output for different inputs.

Test Number	Input	Expected Output	Actual Output
1			
2			
3			

Now try the relational operators below, can you work out what each one does?

>	More than	<=	
<		!=	
==	Equal to		
>=			



Self Assessment:	Exit Ticket: When should you use an if statement?
R A G	

4. Until loop

Correct this code so it works correctly:

```
cartoon = Input("What is your favourite cartoon?")
if carton == "Regular Show":
    print("Oh yeah!)
else
    print(My mum!")
```

```
while True:
    maths = input("What is 5 X 4?")
    if maths==20:
        break
```

Run the code. Try typing in the correct answer, the program should stop but it doesn't can you work out why?

Challenge

Replace the second line with the following code:

```
maths = int(input("What is 5 X 4?"))
```

Run the code again and see what happens this time. Why do you think this is?

Extension

Research what the following functions do and complete the table below.

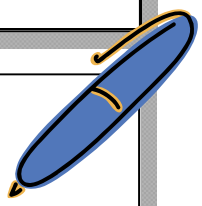
Function	Data type	Example data
str()		
int()		
float()		

Can you write an example program that uses each data type?

Self Assessment: R A G	Exit Ticket: How do you control how many times an until loop repeats?

STRENGTH	TARGET	ACTION	EFFORT

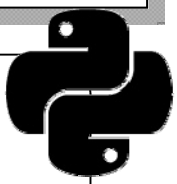
Green Pen Activity:



5. Arithmetic operators

Correct the following code so it runs correctly:

```
while true:
    maths = input("What is 100 X 4?)
    if maths=400:
break
```



```
multiplication = 5 * 3
print(muliplication)
```

What happens when you run the above code in IDLE?

Challenge

Adapt the code and test the following mathematical operators, complete the table once you have worked out what each one does (the last three are tricky!)

Operator	Description	Example
*	Multiplication	5 * 3 = 15
+		
-		
/		
%		
**		
//		

Can you write an example program for each operator?

Extension

Michael Smith runs a business that sells carpet.

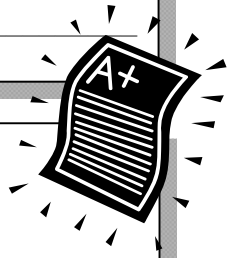
He would like you to write a program that asks the user for the width and the depth of the room and then works out how much carpet they will need.

The program will then ask them what carpet they want to buy (they can choose fromTufted or Woven).

Using the below prices work out how much their choice of carpet will cost in total.

Tufted = £8.00 a meter
Woven = £7.50 a meter

Extend the program to ask if the user wants underlay and add this into the total price for the customer - choose a suitable price per meter for the underlay.



Self Assessment:	Exit Ticket: What is an arithmetic operator?
R A G	

6. Procedures

State what each of these operators do:

+	==
-	!=
/	>
*	<

```
nameOfEmotion = "HAPPY"
def line1():
    print("I am ...")
    for letters in nameOfEmotion:
        print(letters)
line1()
```

What happens when you run the above code in IDLE?

Challenge

Run the code. line1() is a procedure that is used each time you need to print line 1 of the song I am H.A.P.P.Y. When you write line1() at the end of the program this is called calling a procedure.

Try writing a different emotion where you have assigned the variable at the top.
What do you think the following line of code does (hint this is a type of loop):

```
for letters in nameOfEmotion:
```

Extension

The other line of the song is:

I know I am, I'm sure I am

The completed program should run as follows:

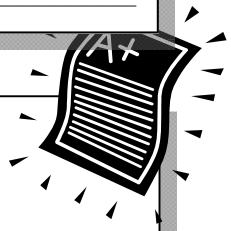
line1()

line1()

line2()

line1()

Can you work out how to define a second procedure for line2() and add this to your program, adapting the call to the functions at the end so the song is printed in the correct order on screen.



Self Assessment:	Exit Ticket: What is a procedure?
R A G	

Keywords

Pseudocode	A description of a computer programming algorithm that uses the structural conventions of a programming language, but is intended for human reading rather than machine reading.
Algorithm	A set of rules specifying a how to solve a problem.
Decision	A selection from a range of options depending upon the result of a condition.
Process	To perform logical operations on (data) according to programmed instructions in order to achieve a desired result.
Function	A subroutine that executes the statements and returns a single value to the program.
Procedure	A subroutine that executes the statements and returns control to the program.
Selection	The pathway through a program is selected by using a condition to decide on what instructions to execute next.
Sequence	Set of instructions to be carried out in the order they are written.
Statement	A single instruction or step within a program.
Subroutine	A subset of code within a larger program, which performs a specific task.
Assignment	Sets or resets the value stored in the storage location denoted by a variable name.
Algorithm	A set of rules specifying a how to solve a problem.
Iteration	A group of instructions is executed repeatedly until a condition is met (a loop).
Nesting	When control structures are inserted within other control structures.
Array	A block of variables of the same type using a single name and an index value.
Boolean	Variables that store just two values, e.g. TRUE or FALSE.
Character	Data type that stores a single character.
Constant	Name used to identify a value in memory that does not change during the execution of the program.
Integer	Whole number values, positive or negative.
Real	Data type that will store decimal (or fractional) values.
String	Data type used to store a string of characters.
Variable	Name used to identify a value in memory that can change during the execution of the program.