Python Programming Challenges

Book 2

Practising Python & Planning Programs using Pseudocode

Name	

Keep a record of the challenges completed here:

Chall. No.	Title/Topic	Completed in class (tick)	Completed as homework (tick)	Any errors? (Yes/No)	How hard did you find this? *(1 being easy, 5 being hard)*	Do you need further practice in this? (Yes/No)
1	Best friend *Variables, inputs & strings*					
2	Favourite song *Variables, inputs & strings*					
3	Piggybank *Type casting, maths operators*					
4	Customer Discount *Type casting, comparison operators*					
5	Piggybank2 *Type casting, comparison operators, selection*					
6	Blast off *Variables, while loop, import functions*					
7	Times tables *Variables, inputs, while loops*					
8	Rolling the dice *Random function, inputs, type casting, selection*					
9	Level 1 Programming assessment					

Instructions:

1. **For each task, you must plan the program using pseudocode BEFORE you create it**
2. **When you have created the program in Python, you must then annotate anything in your pseudocode that was incorrect – use a different colour pen to do this**
3. **Complete a self review for each task:**

Eg:

	Your response
Completed successfully?	Yes or no
Was your pseudocode plan:	Fully correct Mostly correct A bit correct Not at all correct
Did you have any errors?	Yes – syntax errors (write in the error) or No
How did you solve them?	How did you solve your errors/solve the problem? Explain what you did and use examples of code
What did you find easy/difficult?	What was hard/did you struggle with?
Did you work with anyone to complete this challenge?	Yes - write their name No

- **Ask for help from others**
- **If you coach someone else through completing one of the challenges, keep a record of it on the coaching sheet beloe**
- **Complete them in sequence and do not move on to the next challenge until you have successfully got the program running**
- **Make sure that you save your solutions to your Python Practice folder using the challenge title/topic as the file name**
- **Look at the programs that you have completed in previous lessons for help if you struggle**

Coaching record:

Person coached	**Challenge**	**How I helped**

Challenge 2.1: Best friend

Challenge:

Write a program that asks for 2 of your friend's names and then states which friend is the best friend.

Pseudocode:

Self review:

	Your response			
Completed successfully?				
Was your pseudocode plan:	Fully correct	Mostly correct	A bit correct	Not at all correct
Did you have any errors? What?				
How did you solve them?				
What did you find easy/difficult?				
Did you work with anyone to complete this challenge?				

Challenge 2.2: Favourite song

Try the following code:

```
start = "Hello, "
name = input("What is your name? ")
end = ". How are you today?"
sentence = start + name + end
```

Challenge:

Write a program that asks the user what their favourite song is and then asks them for line 1, line 2 and line 3. It should store the first, second and third lines of the song as variables and then add the strings together to output the song.

Pseudocode:

Self review challenge 2.2:

	Your response			
Completed successfully?				
Was your pseudocode plan:	Fully correct	Mostly correct	A bit correct	Not at all correct
Did you have any errors? What?				
How did you solve them?				
What did you find easy/difficult?				
Did you work with anyone to complete this challenge?				

Challenge 2.3: Piggybank

Remember that input() always returns a string. You need to use type casting to convert a string (str) to an integer (int) or integer to a string.

Challenge:

Write a program that helps the user to add up the coins in their piggy bank. The program should ask:
“How many pennies”, “How many two pences”, “How many 5 pences” etc. Then it should give the total value of the piggy bank.

Pseudocode:

Self review, challenge 2.3:

	Your response
Completed successfully?	
Was your pseudocode plan:	Fully correct Mostly correct A bit correct Not at all correct
Did you have any errors? What?	
How did you solve them?	
What did you find easy/difficult?	
Did you work with anyone to complete this challenge?	

Challenge 2.4: Customer Discount

A shop is having a sale. They're giving 10% off when a customer spends £10 or less and 20% off when they spend over £10. Write a program that asks for the amount spent and then displays the discount to be applied and then the final price (ie with the discount applied)

Pseudocode:

Self review, challenge 2.4:

	Your response			
Completed successfully?				
Was your pseudocode plan:	Fully correct	Mostly correct	A bit correct	Not at all correct
Did you have any errors? What?				
How did you solve them?				
What did you find easy/difficult?				
Did you work with anyone to complete this challenge?				

Challenge 2.5: Piggybank 2

Remember that programs can make decisions based on the input of the user – it checks IF a condition is met and IF not it can do something ELSE.

To compare values we need to use comparison operators such as: **< > <= >= !=**

Challenge:

Using your Piggybank program from challenge 2.3; develop the program so that if the users savings add up to less than £50 it tells them to save more and if it is over £50 it tells them that they're doing well.

Pseudocode:

Self review, challenge 2.5:

	Your response
Completed successfully?	
Was your pseudocode plan:	Fully correct Mostly correct A bit correct Not at all correct
Did you have any errors? What?	
How did you solve them?	
What did you find easy/difficult?	
Did you work with anyone to complete this challenge?	

Challenge 2.6: Blast off

Look at the code:

```
number = 1
while number < 101:
     print (number)
     number = number +1
```

Challenge:

Write a program that counts a blast off sequence for a space rocket, counting down from 10 and then saying ‘BLAST OFF’ . You will need to use the import time function and a count variable.

Pseudocode:

Self review, challenge 2.6:

	Your response
Completed successfully?	
Was your pseudo code plan:	Fully correct Mostly correct A bit correct Not at all correct
Did you have any errors? What?	
How did you solve them?	
What did you find easy/difficult?	
Did you work with anyone to complete this challenge?	

Challenge 2.7: Times table

Write a program to print a multiplication table (a times table). At the start it should as the user which number they want to see the times table for by asking “Which times table would you like?”

Hints: you will need to use 2 variables and one of these will be the user input (the times table wanted).

Example output:

```
Which times table would you like?
5
Here’s your table:
5 x 1 = 5……..
```

Pseudocode:

Self review, challenge 2.7:

	Your response			
Completed successfully?				
Was your pseudo code plan:	Fully correct	Mostly correct	A bit correct	Not at all correct
Did you have any errors? What?				
How did you solve them?				
What did you find easy/difficult?				
Did you work with anyone to complete this challenge?				

Challenge 2.8: Rolling the dice

Remember this? Try it:

```
import random
n = random.randint(1,100)
print (n)
```

Challenge:

Write a program that simulates (acts like) rolling a die. The program should ask the user if they want to roll the dice, while they say yes it should tell them the number they have rolled ("You have rolled a "). It should ask them again after each roll. If they say no, the program should say 'Goodbye'

Pseudocode:

Self review, challenge 2.8:

	Your response			
Completed successfully?				
Was your pseudo code plan:	Fully correct	Mostly correct	A bit correct	Not at all correct
Did you have any errors? What?				
How did you solve them?				
What did you find easy/difficult?				
Did you work with anyone to complete this challenge?				

OCR Level 1 Programming Assessment

A company produces games to run on digital television sets. You have been asked to write a quiz program for them. The quiz can be on a theme of your choice. Where you are asked to give evidence of your program you should print out or produce a screenshot of your program. You should check your program works at every stage but only need to give proof of testing in section (i).

Your program will:

- Start by asking for your name.
- Welcome you to the quiz by name.
- Ask five multiple choice questions.
- For each question, if the player has the correct answer, add one to their score.
- At the end of the quiz display the player's name and score .
- Display Well Done 5 times if a player has a score of more than 3.

a) List what your quiz needs to be able to do. Plan the quiz program you are going to make. (You can use flowcharts, pseudo code or any other sensible method of showing how your program will work.)

b) Write a program to ask the player their name and then welcome them to the quiz. For example:

```
a. What is your name? Jerry
b. Hello, Jerry welcome to the cheese quiz
```

Test your program works then give evidence of your code.

c) Add code to your program so asks a question and takes in an answer:
For example:

```
Question One: Which of the following is a goat cheese?
A) Cheddar
B) Caprino
C) Stilton
D) Brie
Enter your answer B
```

Test your program works then give evidence of your code.

d) Improve your program so it adds one to the score if the player gets the answer right.
Test your program works then give evidence of your code.

e) Add four more questions to your quiz.
Test your program works then give evidence of your code.

f) When the quiz is over add code so it prints out the player's score.

```
Jerry has a score of: 5
```

Test your program works then give evidence of your code.

g) If the player has a score of greater than three display: Well Done
Test your program works then give evidence of your code.

h) Change your program so it says “well done” 5 times if the player has a score of greater than 3. Test your program works then give evidence of your code.

i) Test your program works. Make notes on how you tested it and what you found out.

j) Write an evaluation of how well your program works. You should include:

- Any problems you had making it and how you overcame them.
- Any bugs your program still has.
- Any future improvements that could be made to your program.

Self assessment for Level 1 assessment:

	Your response
Completed successfully?	
Was your pseudo code plan:	Fully correct Mostly correct A bit correct Not at all correct
Did you have any errors? What?	
How did you solve them?	
What did you find easy/difficult?	
Did you work with anyone to complete this challenge?	

Glossary

Argument	A piece of information that is required by a function so that it can perform a task
Bug	A piece of code that is causing a program to fail or not to run properly
Comments	Some text in a computer program that is for the human reader and is ignored by the computer
Comparison operators	Also called logic operators. They allow us to compare data. < > <= >= != = =
Data type	Different types of data stored by the computer – for example integers (numbers), text and floats (decimals)
Escape characters	Characters that can be used in Python to 'break' a sequence and tell Python that a character has a different meaning – examples are \ \n (new line) \t (tab indent) \\ (allows a back slash in a string) \" allows speech marks to be used in a string without ending the string
Float	A decimal number
For loop	A loop that repeats code for a specified number of times
Function	A reusable piece of code
Infinite loop	A piece of code that keeps running forever
Integer	A number data type – a whole number
Loop	A piece of code that repeats until a specified condition is met
Mathematical operators	An operator that performs a mathematical calculation, such as + - / * ** %
Module	A saved python file whose functions can be used by another program (eg import time – imports the time module)
Output	Data that is sent from the program to the screen or printer or other output device
Pseudocode	
String	Text data
Syntax	The format of the code
Syntax error	An error produced when a computer fails to run a program because it cannot recognise the format of the code – for example a bracket has not been closed
Testing	Where the program created is run repeatedly using different inputs and conditions to check that it works and runs correctly – every possibility has to be checked.
Type casting	The process of converting a data type to another – eg converting an integer to a string
Variable	A name given to a piece of data that is then stored in the memory and then is used to refer to that data
While loop	A loop that repeats code while a condition is being met (eg while n < 10)