**Python Programming Challenges Book 2**

**Practising Python & Planning Programs using Pseudocode**

|  |  |
| --- | --- |
| **Name** | **Carl Baines** |

1

**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 |

∙

2

**Challenge 2.1: Best friend**

**Challenge:**

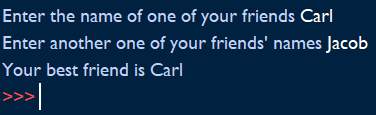
Write a program that asks for 2 of your friend’s names and then states which friend is the best friend. **Pseudocode:**

**user\_input = input("Enter the name of one of your friends ")**

**user\_input2 = input("Enter another one of your friends' names ")**

**bestfriend = user\_input**

**print("Your best friend is " + bestfriend)**



**Self review:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudocode plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it easy to make the code take in user inputs.** |
| **Did you work with anyone to complete this challenge?** | **No** |

3

**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:**

**favsong = input("What is your favourite song? ")**

**artist = input("Who is it by? ")**

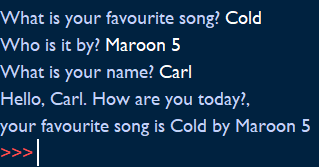
**start = "Hello, "**

**name = input("What is your name? ")**

**end = ". How are you today?,"**

**sentence = start + name + end**

**print(sentence + "\nyour favourite song is " + favsong + " by " + artist)**



4

**Self review challenge 2.2:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudocode plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it easy to take in user inputs and then combine them together in another variable to be printed.** |
| **Did you work with anyone to complete this challenge?** | **No** |

5

**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:**

**pennies = int(input("How many pennies are in your piggy bank? "))**

**twop = int(input("How many two pences are in your piggy bank? "))**

**fivep = int(input("How many five pences are in your piggy bank? "))**

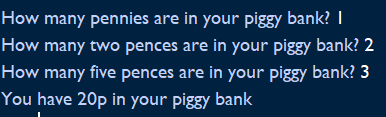
**ptotal = pennies**

**twototal = twop\*2**

**fivetotal = fivep\*5**

**total = ptotal + twototal + fivetotal**

**print("You have " + str(total) + "p in your piggy bank")**



6

**Self review, challenge 2.3:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudocode plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it easy converting an integer to a string.** |
| **Did you work with anyone to complete this challenge?** | **No** |

7

**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:**

**amountspent = int(input("How much did you spend in pounds? "))**

**if amountspent <= 10:**

**discount = 1.10**

**print("Your discount is 10% off")**

**finalprice1 = amountspent/1.1**

**print("You need to pay £" + str(finalprice1))**

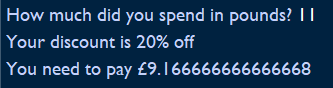
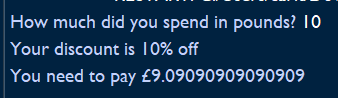
**else:**

**discount = 1.20**

**print("Your discount is 20% off")**

**finalprice2 = amountspent/1.2**

**print("You need to pay £" + str(finalprice2))**



8

**Self review, challenge 2.4:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudocode plan:** | Mostly correct |
| **Did you have any errors? What?** | **Logic error – my total prints with lots of decimals on the end.** |
| **How did you solve them?** | **I don’t know how to solve it.** |
| **What did you find**  **easy/difficult?** | **I found it easy to take a user input and calculate the final price by applying the discount.** |
| **Did you work with anyone to complete this challenge?** | **No** |

9

**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:**

**pennies = int(input("How many pennies are in your piggy bank? "))**

**twop = int(input("How many two pences are in your piggy bank? "))**

**fivep = int(input("How many five pences are in your piggy bank? "))**

**ptotal = pennies**

**twototal = twop\*2**

**fivetotal = fivep\*5**

**total = ptotal + twototal + fivetotal**

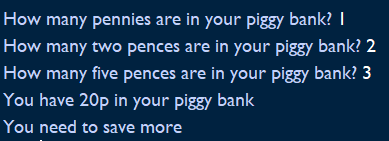
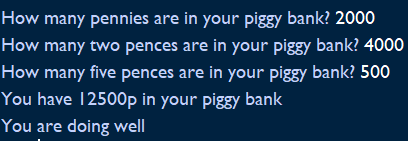
**print("You have " + str(total) + "p in your piggy bank")**

**if total <5000: #£50 in pennies**

**print("You need to save more")**

**else:**

**print("You are doing well")**



10

**Self review, challenge 2.5:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudocode plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it difficult converting the pennies into pounds if it was needed so I decided to leave it out of my code.** |
| **Did you work with anyone to complete this challenge?** | **No** |

11

**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:**

**import time**

**number = 5**

**count = 0**

**while number < 6:**

**time.sleep(0.2)**

**print (number)**

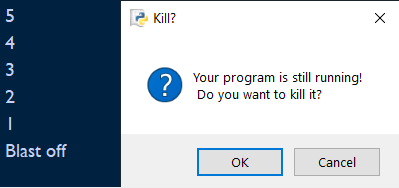
**number = number - 1**

**count = count + 1**

**while count == 5:**

**print("Blast off")**

**quit()**



12

**Self review, challenge 2.6:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudo code plan:** | Fully correct |
| **Did you have any errors? What?** | **Yes – when the number variable went below 0 it would print blast off but then start printing minus numbers.** |
| **How did you solve them?** | **I put the quit () function in after blast off had been printed to prevent it.** |
| **What did you find**  **easy/difficult?** | **I found adjusting the while loop, in the example, to meet with the requirements of the code easy.** |
| **Did you work with anyone to complete this challenge?** | **No** |

13

**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:**

**import time**

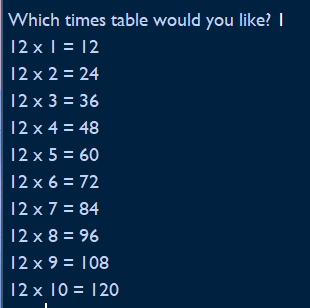
**num = 12**

**timestable = int(input("Which times table would you like? "))**

**for i in range(1,11):**

**time.sleep(0.2)**

**print(num, "x", i, "=", num\*i)**



14

**Self review, challenge 2.7:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudo code plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it easy generating all the times tables for the user input.** |
| **Did you work with anyone to complete this challenge?** | **No** |

15

**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:**

**import random**

**def sim():**

**start = input("Do you want to roll the dice? ")**

**if start == "yes" or start == "y":**

**n = random.randint(1,100)**

**print ("You have rolled a " + str(n))**

**else:**

**print("Goodbye")**

**quit()**

**rollagain = input("Do you want to roll again? ")**

**if rollagain == "yes" or rollagain == "y":**

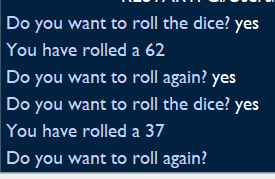
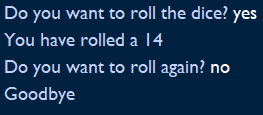
**sim()**

**else:**

**print("Goodbye")**

**quit()**

**sim()**



16

**Self review, challenge 2.8:**

|  |  |
| --- | --- |
|  | **Your response** |
| **Completed successfully?** | **Yes** |
| **Was your pseudo code plan:** | Fully correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found it easier by using a function and then calling it if the user replies with yes to the ‘rollagain’ variable.** |
| **Did you work with anyone to complete this challenge?** | **No** |

17

**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.

1. 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.)

My program needs to be able to take the user’s name as an input and then use the input to welcome them to the quiz. The code should then ask the user 5 multiple choice questions and each time the user gets it right, the code should display a message and add one to the score. If the user gets it incorrect, the code should also print a message. At the end of the quiz, the final score should be calculated and outputted.

Psuedocode:

import time

playerscore = 0

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

time.sleep(0.5)

print("Welcome to the football quiz, " + name)

q1 = input("Question One: Which football team has won the most premier leagues?\n A) Liverpool\n B) Arsenal\n C) Manchester United\n D) Manchester City ")

if q1 == "C" or q1 == "c":

time.sleep(0.2)

print("That is the correct answer! You got a point!")

playerscore = playerscore + 1

else:

print("That is the incorrect answer :(")

q2 = input("Question Two: Who scored the most goals in Premier League History?\n A) Alan Shearer\n B) Harry Kane\n C) Sergio Auguero\n D) Thierry Henry ")

if q2 == "a" or q2 == "A":

time.sleep(0.2)

print("That is the correct answer! You got a point!")

playerscore = playerscore + 1

else:

print("That is the incorrect answer :(")

q3 = input("Question Three: Who were the first club to win the Premier League?\n A) Blackburn\n B) Manchester United\n C) Liverpool\n D) Spurs ")

if q3 == "B" or q3 == "b":

time.sleep(0.2)

print("That is the correct answer! You got a point!")

playerscore = playerscore + 1

else:

print("That is the incorrect answer :(")

q4 = input("Question Four: Who scored the fastest ever Premier League goal?\n A) Wayne Rooney\n B) Teddy Sheringham\n C) Sadio Mane\n D) Shane Long ")

if q4 == "D" or q4 == "D":

time.sleep(0.2)

print("That is the correct answer! You got a point!")

playerscore = playerscore + 1

else:

print("That is the incorrect answer :(")

q5 = input("Question Five: Who has made the most appearances in the Premier League?\n A) Gareth Barry\n B) Frank Lampard\n C) James Milner\n D) Wayne Rooney ")

if q5 == "a" or q5 == "A":

time.sleep(0.2)

print("That is the correct answer! You got a point!")

playerscore = playerscore + 1

else:

print("That is the incorrect answer :(")

time.sleep(0.5)

print(str(name) + " scored " + str(playerscore) + " out of 5")

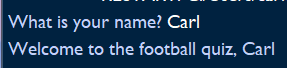
if playerscore > 3:

print("Well Done!\nWell Done!\nWell Done!\nWell Done!\nWell Done!")

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

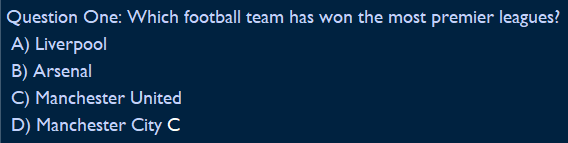
B) Caprino

C) Stilton

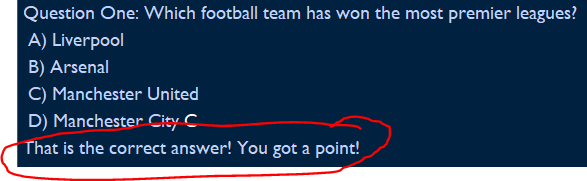
D) Brie

Enter your answer B

Test your program works then give evidence of your code.

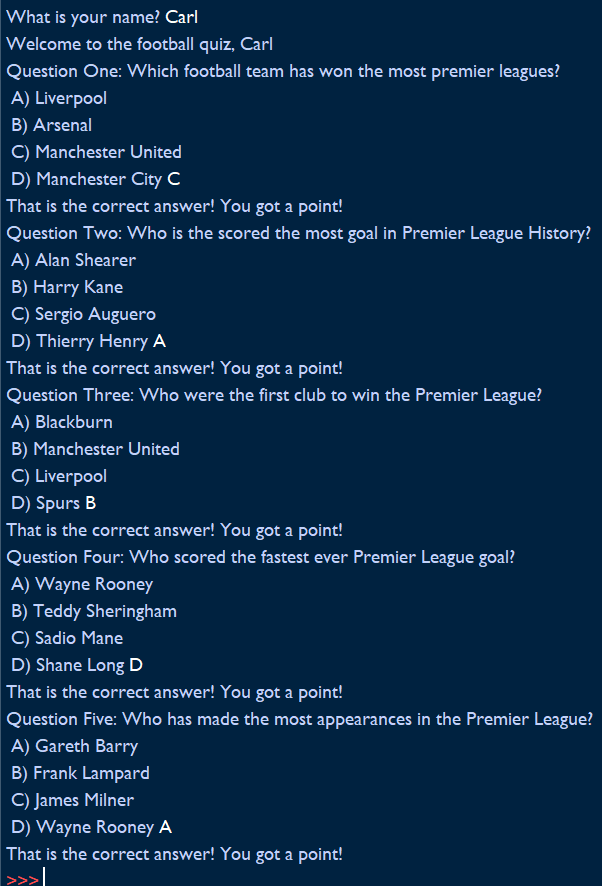
d

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.



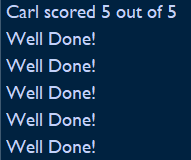
g) If the player has a score of greater than three display: Well Done

Test your program works then give evidence of your code.



18

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?** | **Yes** |
| **Was your pseudo code plan:** | Fully correct Mostly correct A bit correct Not at all correct |
| **Did you have any errors? What?** | **No** |
| **How did you solve them?** | **N/A** |
| **What did you find**  **easy/difficult?** | **I found the whole code easy.** |
| **Did you work with anyone to complete this challenge?** | **No** |

19

**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) |

20