Question **1**

Correct

Mark 1 out of 1

In designing control algorithms for an autonomous vehicle, which feature is essential for ensuring safety and reliability?

- ⦿ a. The ability to process data from multiple sensors and make real-time decisions ⊘
- ○ b. The number of users that can interact with the system and make decisions
- ○ c. The length of the control code to enable and evaluate safety features in real time
- ○ d. The use of complex mathematical equations to simulate almost any evantuality

The correct answer is: The ability to process data from multiple sensors and make real-time decisions

Question **2**

Correct

Mark 1 out of 1

What is the primary benefit of using version control in collaborative software development?

- ○ a. It ensures that all developers write code in the same style
- ○ b. It eliminates the need for testing or instrinsic documentation
- ○ c. It makes the code less readable by allowing multiple programmers to contribute to a solution
- ⦿ d. It allows for the tracking of changes and management of different versions of the code ⊘

The correct answer is: It allows for the tracking of changes and management of different versions of the code

Question 3

Incorrect

Mark 0 out of 1

What is the purpose of the requirements definition step in software development?

- ○ a.   To create the final software product
- ○ b.   To gather and clarify the needs of stakeholders
- ○ c.   To determine the cost of the project
- ◉ d.   To outline the programming languages to be used ⊗

The correct answer is: To gather and clarify the needs of stakeholders

Question 4

Correct

Mark 1 out of 1

Which of the following approaches is most beneficial when optimising code for performance?

- ◉ a.   Refactoring code to eliminate redundancy and improve clarity while enhancing efficiency ⊘
- ○ b.   Writing more comments to explain the code
- ○ c.   Increasing the size of the main function to include all logic and reduce the need for add-ons and plug-ins.
- ○ d.   Reducing the number of variables used in the code o make exection as fast as possible

The correct answer is: Refactoring code to eliminate redundancy and improve clarity while enhancing efficiency

desk check this algorithm with values 2, 3, 0, 1

```
Start
    get a, b, sum, product
    For i from 1 to b
        sum = sum + a
        product = product * i
        Display sum, product
    Next i
End
```

a	b	sum	product	i	display
2	3	0	1		
				1	
		2			
			1		
					2,1
				2	
		4			
		2			
					4,2
				3	

Marking guide

Mark	Content
3	Correct desk check with all values, calculations and outputs noted
2	OArtially correct desk check
1	provides some relevant information

Sample answer

a	b	sum	product	i	Display
2	3	0	1	1	
		2	1		2, 1
				2	

		4	2		4, 2
				3	
		6	6		6, 6

Comment:

Drag the correct lines into this algorithm so it will find the average and mark and maximm mark from a data file called MarksData

```
BEGIN Analysis

        Sum = 0

        Max = 0

        Num_Students = 0

        Open MarksData for input

        Read Mark from MarksData

        WHILE NOT EOF

                  IF Mark > Max THEN      ⊘

                        Max = Mark

                  ENDIF

                  Add 1 to Num_Students    ⊗

                  Sum = Sum + Mark         ⊗

              Average = Sum/Num_Students

              Read Mark from MarksData

                  END WHILE              ⊘

        Display Max, Average

        Close MarksData

END Analysis
```

Add 1 to Num_Students	END WHILE	Sum = Sum * Mark	Num_Students = Max	IF Mark > Max THEN
Sum = Sum + Mark	END	END IF	IF Mark < Max THEN	

Your answer is correct.

The correct answer is:
Drag the correct lines into this algorithm so it will find the average and mark and maximm mark from a data file called MarksData

```
BEGIN Analysis

        Sum = 0

        Max = 0

        Num_Students = 0

        Open MarksData for input

        Read Mark from MarksData

        WHILE NOT EOF

                [IF Mark > Max THEN]

                      Max = Mark

                ENDIF

                [Sum = Sum + Mark]

                [Add 1 to Num_Students]

                Average = Sum/Num_Students

                Read Mark from MarksData
```

```
        [END WHILE]
        Display Max, Average
        Close MarksData
END Analysis
```

Comment:

A palindrome is a word that reads the same forward and backward, such as the words 'refer' and 'noon'.

The following code fragment checks whether a word is a palindrome.

```
BEGIN PalindromeCheck(word, palindrome)

     palindrome = TRUE

     F = 1

     N = number of letters in word

     L = N

     REPEAT

          IF Fth letter of word is not equal to Lth letter of word THEN

               palindrome = FALSE

          END IF

          L = L-1

          F = F+1

     UNTIL F > N

     RETURN palindrome

END PalindromeCheck
```

Explain the effect of changing the 12th line of code from `UNTIL F > N` to `UNTIL F > N/2 OR palindrome = FALSE`.

The effect of changing line 12 will cause the code to stop if the condition "f letter of word is not equal to L letter of word" causing the program to end the loop before F > N making the code not function as intended.

Marking Guide

Mark	Content
3	Provides correct explanations for BOTH modifications
2	Provides a correct explanation for ONE modification
1	Demonstrates some understanding of the algorithm

Sample answer

UNTIL F>N/2 – once the process has reached halfway, the same checks would be repeated unnecessarily.

Adding 'OR palindrome = FALSE' – there is no need to keep checking once a non-matching letter is found.

In both cases, the loop can be exited early, improving efficiency.

Comment:

For a particular concert, there are 10 000 tickets available, and 2000 customers wanting to buy tickets. Customers are allocated tickets randomly.

Customers are required to indicate the number of tickets they would like to purchase, up to a limit of 8.

Seats are not allocated until all ticket requests have been entered.

Details of the 2000 customers are already stored, in no particular order, in an array of records called Customer_Array.

Each customer record has the following structure:

Cust_ID : String

Name : String

Num_Tickets : Integer

Allocated : Boolean (becomes TRUE when tickets have been allocated)

For example:

Cust_ID	Name	Num_Tickets	Allocated
Ervine123	Ervine, D	1	FALSE
John117	Chief, M	3	FALSE

An algorithm called AllocateTickets is required to allocate ticket numbers to customers.

Customers are selected at random to receive their requested tickets until all available tickets are allocated. The tickets allocated will always be sequentially numbered.

The Cust_ID of the chosen customers are stored next to the relevant ticket number in a 10 000-element array of records called Ticket_Array. For example:

TicketNumber	AllocatedCustomer
1	Matt72
2	Matt72
3	Jeff72
4	Jeff72
...	...
9998	Anil123
9999	Leah80
10000	Leah80

As the last tickets are allocated, a customer may not get all the tickets they have requested (as there may be no more tickets left). Some customers may not be allocated any tickets at all.

In the example shown above, even if *Leah80* requested five tickets, there will only be two allocated because that is all that is left.

Design an appropriate algorithm for AllocateTickets.

You may use this function:

Rand (Min, Max) generates a random integer between Min and Max, inclusive

BEGIN

  get Cust_id

  get name

```
    get Num_tickets
   Allocated = False
   tickets_allocated_list = []
   FOR customer from 1 to 10000
     ticket_number = rand(1,10000)
     add tick_number to ticket allocated list
      IF ticket_number is IN ticket_allocated_list THEN
        skip ticket number
     ENDIF
     Allocated = TrueAllocated = True
   NEXT customer
   display ticket_number , name
  END
```

```
      FOR i = 1 to Num_Tickets

            Ticket_Array[TixIndex].AllocatedCustomer = CustId

            Add 1 to TixIndex

      NEXT i

END Allocate
```

Comment:
you get Num_tickets but then don't use it.

Use this stimulus to answer both questions on this page.

A software documentation system is devised that automatically produces different details of documentation depending on the intended use of the software.

The following algorithm calculates the total cost (TC) of a number of concert tickets (N), allowing for discounts (D) for certain types of customers (C). *(Line numbers are added for reference only)*

```
1 BEGIN program
2 input N
3 input C
4 if C = "Member" then
5 D=0.25
6 TC= N * 30 *(1-D)
7 display TC
8 else
9 if C = "Student" then
10 D=0.10
11 TC= N * 30 *(1-D)
12 display TC
13 else
14 TC =N * 30
15 display TC
16 end if
17 end if
18 END program
```

Create a data dictionary for this algorithm

variable	value	use	works
N	input from user	gets the number of tickets	yes
c	input from user	gets the type of customer	yes
d	0.25	applies a 25% off discount to ticket if client is a member	yes
TC	value is selection based if member ; N *30 * (1-D)	TC variable is the total cost of the ticket and is determined based on the client membership if they are member,student,regular	yes

Marking guide

Mark	Content
4	Complete data dictioanry for all variables used in the algorithm. Answers in this range will include all relevat metadata about variables in the table.

3	Mostly complete data dictioanry for all variables used in the algorithm
2	Partially complete data dictionary for the algorithm
1	Provides some relevant information

Comment:

Question **10**

Complete

Mark 2 out of 4

Describe ways in which the algorithm could be improved so that it is easier to maintain?

Instead of using a long chain of if statements the algorithm could use CASEWHERE statements to make reading and selection of discount more efficient and maintainable. When looking at the algorithm I struggled to keep track of value of discount due to the chain of if statements so by using switch statements the algorithm would be improved and easy to read/maintain.

Mark	Content
4	Describes a number of relevant measures that could be taken to improve maintainablility of the algorithm Answers in this reange will provide comprehensive examples to support the response.
3	Describes a number of relevant measures that could be taken to improve maintainablility of the algorithm
2	Outlines a relevant measure to improve maintainablility of the algorithm
1	Provides some relevant information

Comment:
What about simple things like indents or relevant variable names? I struggled because the algorithm had no valid structure.

I don't disagree with what you've said, but again, none of it will make sense without the use of real variable names, and proper indents.

What is the purpose of using breakpoints when debugging software solutions?

○ a. To pause the execution of code at a specific line for inspection ⊘

○ b. To display the final output of the program once compiled

○ c. To automatically fix errors in the code as it is being compiled

○ d. To measure the performance of the code at different points of execution

The correct answer is: To pause the execution of code at a specific line for inspection

A robotic system uses a motion sensor to detect obstacles in its path. What is the primary role of this sensor within the system?

○ a. To convert the motion of the robot into code to enable free movement

○ b. To provide real-time data for navigation and decision-making ⊘

○ c. To control the power supply to the actuatorand therefore manage the speed of the robot

○ d. To optimise the efficiency of the actuator

The correct answer is: To provide real-time data for navigation and decision-making

Complete the Binary Number Line

| 128 ✓ | 64 ✓ | 32 ✓ | 16 ✓ | 8 ✓ | 4 ✓ | 2 ✓ | **1** |

Complete the Hexadecimal to Binary Table

Bin	Hex		Bin	Hex
0000	0 ✓		1000 ✓	8 ✓
0001 ✓	1 ✓		1001 ✓	9 ✓
0010 ✓	2 ✓		1010 ✓	A ✓
0011 ✓	3 ✓		1011 ✓	B ✓
0100 ✓	4 ✓		1100 ✓	C ✓
0101 ✓	5 ✓		1101 ✓	D ✓
0110 ✓	6 ✓		1110 ✓	E ✓
0111 ✓	7 ✓		1111 ✓	F ✓

Complete the table of Binary to Decimal to Hexadecimal conversions.

(You can use your answer booklet to help you with your working out if you need.)

Binary	Decimal	Hexadecimal
10110001	167 ✗	B1 ✓
10110011 ✗	**149**	BC ✗
11111101 ✓	251 ✗	**FD**

| 1 | 4 | 128 | 2 | 64 | 6 | 32 | 8 | 16 |

| 4 | C | 1011 | 0101 | 1000 | F | 10 | E | B | 5 | 1 | 1110 | 9 | 1111 | 0000 | 2 | 1100 | G | 0010 | A |

| D | 1001 | 0001 | 0111 | 7 | 0 | 0011 | 6 | 0100 | 1101 | 0110 | 3 | 1010 | 8 |

| 177 | 139 | 93 | 10110011 | 10110011 | BC | 11101000 | 251 | FB | B1 | FE | 11111101 |

| 139 | 253 | 10010011 | 167 | D1 |

Your answer is partially correct.

You have correctly selected 40.

The correct answer is:

Complete the Binary Number Line

| [128] | [64] | [32] | [16] | [8] | [4] | [2] | **1** |

Complete the Hexadecimal to Binary Table

Bin	Hex		Bin	Hex

0000	[0]		[1000]	[8]
[0001]	[1]		[1001]	[9]
[0010]	[2]		[1010]	[A]
[0011]	[3]		[1011]	[B]
[0100]	[4]		[1100]	[C]
[0101]	[5]		[1101]	[D]
[0110]	[6]		[1110]	[E]
[0111]	[7]		[1111]	[F]

Complete the table of Binary to Decimal to Hexadecimal conversions.

(You can use your answer booklet to help you with your working out if you need.)

Binary	Decimal	Hexadecimal
10110001	[177]	[B1]
[10010011]	**149**	[93]
[11111101]	[253]	**FD**
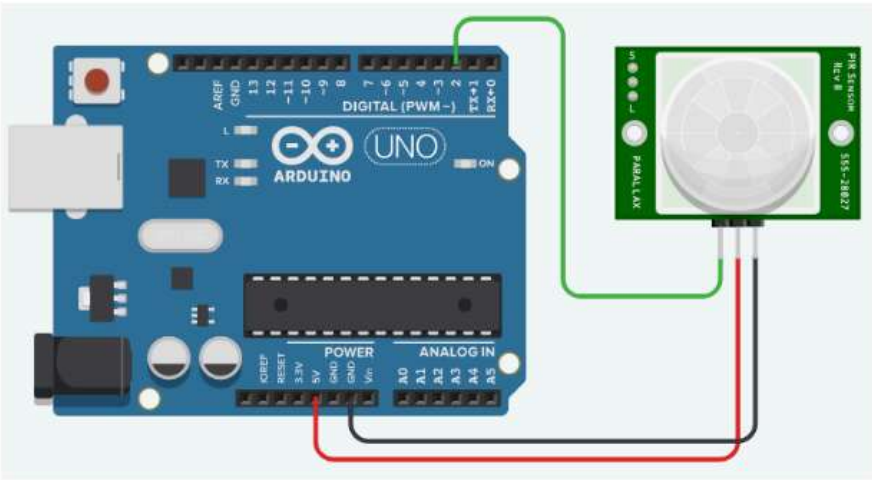
Comment:

Drag and Drop the labels describing the purpose of each wire in the diagram.

Red wire [Power]

Green wire [Analog data]

Black wire [Earth]

[Earth] [Analog data] [Digital data] [Power]

Your answer is partially correct.

You have correctly selected 2.

In a flowchart representing an algorithm, what does a diamond-shaped symbol typically represent?

- ○ a. A decision ⊘
- ○ b. Start or end of the process
- ○ c. Data input or output
- ○ d. A process or action step

The correct answer is: A decision

Match the following terms to their definitions

Binary representations of individual instructions to carry out specific tasks like reading sensor data, performing calculations, or activating actuators

Opcodes ✓

This device interprets program commands, processes data, and coordinates the responses of various subsystems to achieve desired mechanical functions.

CPU ✓

Small, fast storage units that temporarily hold data being processed or transferred.

Address ✗

A complete collection of commands to be understand and executed.

Instruction set ✓

Compact integrated circuits that combine a central processing unit (CPU), memory, and input/output peripherals on a single chip.

Microcontroller ✓

Your answer is partially correct.

You have correctly selected 4.

The correct answer is: Binary representations of individual instructions to carry out specific tasks like reading sensor data, performing calculations, or activating actuators → Opcodes, This device interprets program commands, processes data, and coordinates the responses of various subsystems to achieve desired mechanical functions. → CPU, Small, fast storage units that temporarily hold data being processed or transferred. → Data Registers, A complete collection of commands to be understand and executed. → Instruction set, Compact integrated circuits that combine a central processing unit (CPU), memory, and input/output peripherals on a single chip. → Microcontroller

Question **17**

Correct

Mark 1 out of 1

What is an advantage of using Agile project management over the Waterfall model in software development?

  ◉ a.   Agile allows for more flexibility and iterative improvements ✓

  ○ b.   Agile is less resource-intensive

  ○ c.   Agile requires less planning and documentation

  ○ d.   Agile guarantees a faster delivery of the final product

The correct answer is: Agile allows for more flexibility and iterative improvements

Drag and drop the correct labels onto the (level 1) dataflow diagram

An online medical database stores the medical history of patients. It is updated each time a person seeks medical treatment.

To provide increased security for the database, two-step authentication is implemented which requires a user to:

- provide a valid username and password
- enter an automatically generated six-digit verification code which is sent to their mobile phone.

Complete the dataflow diagram for this two-step authentication process:
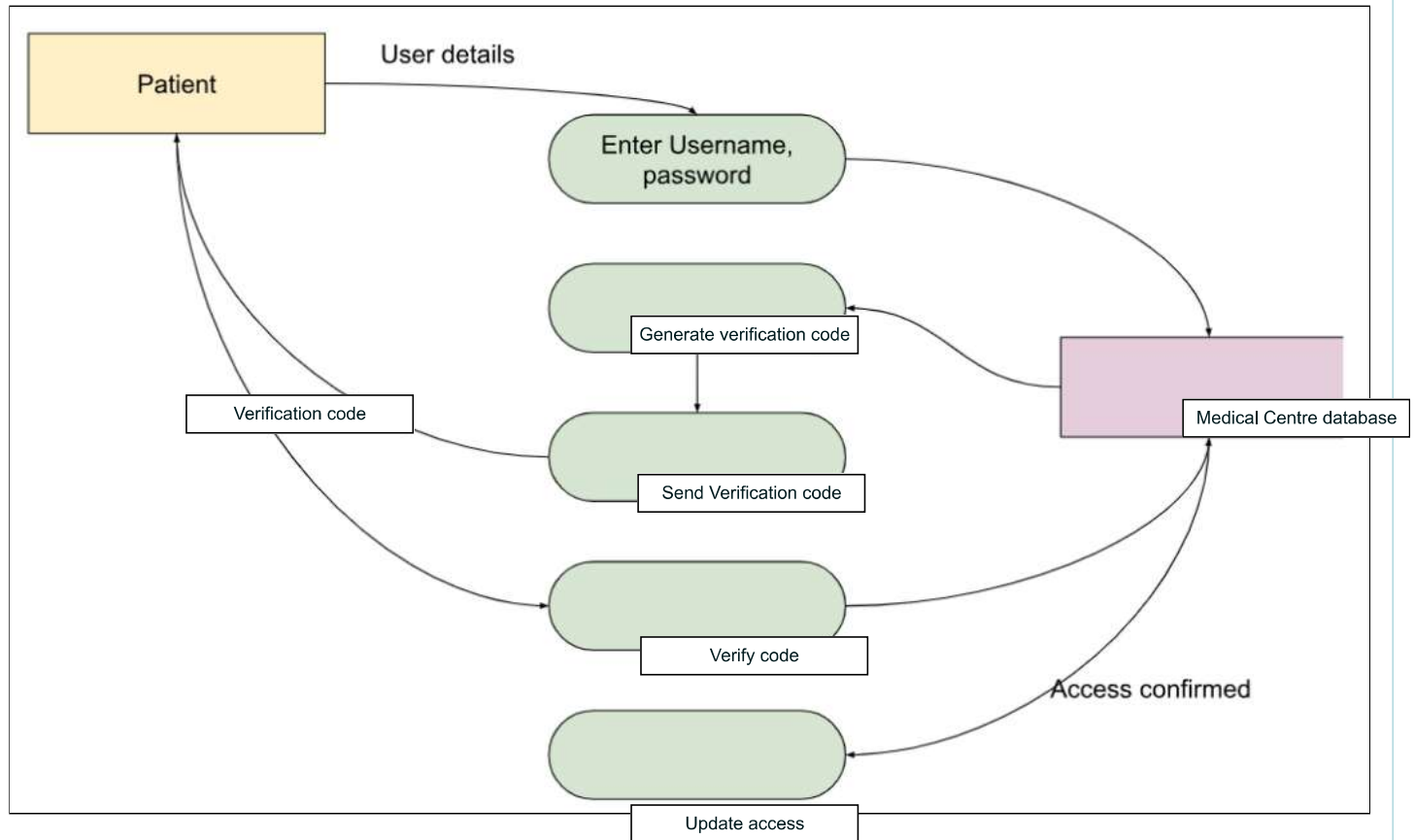
Your answer is correct.

With apologies folks, Coderunner kept crashing my server, so I have reverted to plain to make the coding questions. I apologies for the impact this will have on yoru coding. I won't be mean to you...

Question:

Write a Python program that acts as a simple calculator.

The program should be able to perform addition, subtraction, multiplication, and division.

The user should be prompted to enter **two** numbers and choose the operation.

num1 = int(input("number 1: ")) #get num 1 from user

num2 = int(input("number 2: ")) #get num 2 from user

choice = input("enter operation (add,subtract,divide,multiple): ").lower() #get choice of operation from user

#selection (perform operation based on user input)

if choice == "add":

  print(num1+num2)

elif choice == "subtract":

  print(num1-num2)

elif choice == "divide":

  print(num1/num2)

elif choice == "multiple":

  print(num1*num2)

else: #if invalid input print this

  print("invalid input")

Marking Criteria

1 mark - Use of meaningful variable names

1 mark - print operations

1 mark -indentation

1 mark - correct syntax of if statements

1 mark - Identifies division by zero as an error

1 mark - print result

1 mark - print Invalid input

Sample Answer

```
def calculator():
    print("Simple Calculator")
    print("Select operation:")
    print("1. Addition")
    print("2. Subtraction")
```

```python
    print("3. Multiplication")

    print("4. Division")

    choice = input("Enter choice (1/2/3/4): ")

    if choice in ['1', '2', '3', '4']:

        num1 = float(input("Enter first number: "))

        num2 = float(input("Enter second number: "))

        if choice == '1':

            result = num1 + num2

            operation = 'Addition'

        elif choice == '2':

            result = num1 - num2

            operation = 'Subtraction'

        elif choice == '3':

            result = num1 * num2

            operation = 'Multiplication'

        elif choice == '4':

            if num2 != 0:

                result = num1 / num2

                operation = 'Division'

            else:

                return "Error! Division by zero."

        return f"The result of {operation} is: {result}"

    else:

        return "Invalid input."
print(calculator())
```

Comment:

Question **20**

Incorrect

Mark 0 out of 1

In a scenario where multiple classes share common features, which OOP principle allows for the creation of a base class to encapsulate those features?

- ○ a.   Generalisation
- ○ b.   Polymorphism
- ● c.   Abstraction ⊗
- ○ d.   Inheritance

The correct answer is: Inheritance

Question 21

Correct

Mark 1 out of 1

Which of the following best describes a "class" in object oriented programming?

- a. A data structure for storing multiple values including strings, records and lists
- b. A blueprint for creating objects that encapsulates data and behaviour ⊘
- c. A procedure that performs a specific task that can be used as the basis for amny other tasks
- d. A method for executing code in sequence

The correct answer is: A blueprint for creating objects that encapsulates data and behaviour

Question 22

Correct

Mark 1 out of 1

What is the purpose of using the façade pattern? in software development?

- a. To simplify interactions between a complex subsystem and its clients by providing a unified interface ⊘
- b. To enhance the performance of the code by optimising algorithms to complete tasks in any order
- c. To implement inheritance in a more manageable way by defining every possible iteration of an object
- d. To break down a system into smaller, more manageable components which can then be used as building block for other functions

The correct answer is: To simplify interactions between a complex subsystem and its clients by providing a unified interface

Match the terms to their definitions

This allows objects to be treated as instances of their parent class, enabling a unified interface for different data types. This is often achieved through method overriding and interfaces, allowing for flexibility and the ability to define methods in different ways for different data types.

Inheritance ⊗

This technique allows programmers to focus on the essential qualities of an object, hiding the complex implementation details. This simplifies problem-solving by enabling developers to work with high-level representations.

Abstraction ⊘

This principle involves bundling the data (attributes) and methods (functions) that operate on the data into a single unit, or class.

Generalisation ⊗

This a mechanism where a new class, called a subclass, is created from an existing class, known as a superclass. The subclass inherits attributes and methods from the superclass, allowing for code reuse and the creation of hierarchical relationships.

Polymorphism ⊗

Your answer is partially correct.

You have correctly selected 1.

The correct answer is: This allows objects to be treated as instances of their parent class, enabling a unified interface for different data types. This is often achieved through method overriding and interfaces, allowing for flexibility and the ability to define methods in different ways for different data types. → Polymorphism, This technique allows programmers to focus on the essential qualities of an object, hiding the complex implementation details. This simplifies problem-solving by enabling developers to work with high-level representations. → Abstraction, This principle involves bundling the data (attributes) and methods (functions) that operate on the data into a single unit, or class. → Encapsulation, This a mechanism where a new class, called a subclass, is created from an existing class, known as a superclass. The subclass inherits attributes and methods from the superclass, allowing for code reuse and the creation of hierarchical relationships. → Inheritance

Apologies again folks, Coderunner is crashing my server.

A basic facade that allows a user to enter music into a simple program has been started for you, but is missing the input and display functionality.

Add code to allow for user input of their favourite music. Once entered, the program should display the entered song details.

You will need to add Artist, Song title, Album and song duration (in minutes) to your inputs, and have the input store its data within the facade pattern. It is expected that you will start to code from the #Client Code section of the facade.

The output should show that the song has been entered by reading the facade.

For example, If I add the song Rosanna by Toto (from the album Toto IV), the output should read:

Added Rosanna to the album 'Toto IV'.

(hint: use this example to test your code)

(another hint: comment your code!)

```python
# Song class to represent a song
class Song:
    def __init__(self, title, duration):
        self.title = title
        self.duration = duration  # Duration in minutes
    def __repr__(self):
        return f"Song(title='{self.title}', duration={self.duration})"
# Album class to represent an album that contains songs
class Album:
    def __init__(self, name, artist):
        self.name = name
        self.artist = artist
        self.songs = []
    def add_song(self, song):
        self.songs.append(song)
        print(f"Added {song.title} to the album '{self.name}'.")
    def __repr__(self):
        return f"Album(name='{self.name}', artist='{self.artist}', songs={self.songs})"
# Artist class to represent an artist
class Artist:
    def __init__(self, name):
        self.name = name
        self.albums = []
    def add_album(self, album):
        self.albums.append(album)
    def __repr__(self):
        return f"Artist(name='{self.name}', albums={self.albums})"
# Facade Class
class MusicCollectionFacade:
    def __init__(self):
        self.artists = []
```

```python
        def add_song(self, artist_name, album_name, song_title, duration):
            # Find or create artist
            artist = next((a for a in self.artists if a.name == artist_name), None)
            if artist is None:
                artist = Artist(artist_name)
                self.artists.append(artist)
            # Find or create album
            album = next((a for a in artist.albums if a.name == album_name), None)
            if album is None:
                album = Album(album_name, artist_name)
                artist.add_album(album)
            # Add the song to the album
            song = Song(song_title, duration)
            album.add_song(song)
        def display_collection(self):
            for artist in self.artists:
                print(artist)
# Client Code
class Facade:
    def __init__(self):
        #get information about the song from user (name,duration,album name)
        self.song_name = input("enter song name")
        self.song_duration = input("enter duration in minutes")
        client_song = Song(self.song_name, self.song_duration)
        album_name = input("enter album name")
        #print the song details
        print(client_song)
        print(album_name)
Facade() #call the facade for testing
```

#no code runner is pain

Marking Guide

Mark	Expected Content
5	Working solution that adds songs to the facade and displays appropriate output. Answers in this band will use relevant variable and object names and will comment code appropriately
4	Mostly working solution that adds songs to the facade and displays appropriate output. Answers in this band will use relevant variable and object names and will comment code appropriately
3	Partially working solution that attempts to add and store data to the facade
2	Attempts a solution that tries to use the facade to enter and store data

1	Attempts to code a solution

Sample Answer

```python
# Song class to represent a song

class Song:

    def __init__(self, title, duration):

        self.title = title

        self.duration = duration  # Duration in minutes


    def __repr__(self):

        return f"Song(title='{self.title}', duration={self.duration})"


# Album class to represent an album that contains songs

class Album:

    def __init__(self, name, artist):

        self.name = name

        self.artist = artist

        self.songs = []


    def add_song(self, song):

        self.songs.append(song)

        print(f"Added {song.title} to the album '{self.name}'.")


    def __repr__(self):

        return f"Album(name='{self.name}', artist='{self.artist}', songs={self.songs})"


# Artist class to represent an artist

class Artist:

    def __init__(self, name):

        self.name = name

        self.albums = []


    def add_album(self, album):

        self.albums.append(album)


    def __repr__(self):

        return f"Artist(name='{self.name}', albums={self.albums})"


# Facade Class

class MusicCollectionFacade:

    def __init__(self):

        self.artists = []


    def add_song(self, artist_name, album_name, song_title, duration):

        # Find or create artist

        artist = next((a for a in self.artists if a.name == artist_name), None)
```

```python
            if artist is None:
                artist = Artist(artist_name)
                self.artists.append(artist)


            # Find or create album
            album = next((a for a in artist.albums if a.name == album_name), None)
            if album is None:
                album = Album(album_name, artist_name)
                artist.add_album(album)


            # Add the song to the album
            song = Song(song_title, duration)
            album.add_song(song)


    def display_collection(self):
        for artist in self.artists:
            print(artist)


# Client Code
if __name__ == "__main__":
    music_collection = MusicCollectionFacade()


    # Input system to add songs
    while True:
        artist_name = input("Enter artist name (or 'exit' to quit): ")
        if artist_name.lower() == 'exit':
            break
        album_name = input("Enter album name: ")
        song_title = input("Enter song title: ")
        duration = float(input("Enter song duration (in minutes): "))


        music_collection.add_song(artist_name, album_name, song_title, duration)


    # Display the music collection
    print("\nMusic Collection:")
    music_collection.display_collection()
```

Comment:
I agree about no coderunner = pain, but no server = bigger pain, so I had to compromise.