Assignment 3

Code Review

Done by:

Elaf Abdullah Saleh Alhaddad 31063977
Sutulova Tatiana 30806151

1. How review meetings are conducted

Since our team consists of two people, it was simple to divide the workload: while creating the calendar app we decided that each teammate will be responsible for implementing several functionalities and therefore we had the equal number of methods/functions to do. As for reviewing the code, we have conducted 3 code review meetings : in the first meeting, one team member had a role of reviewer and the other one, the actual creator of functions being reviewed, had a role of developer, whereas in the second review the roles were swapped. Due to the CMCO and the global pandemic the code review meetings were conducted virtually through zoom.

The reviewer shares their screen with the code being reviewed. Using the review checklist, they state what the weak and strong parts of the code are, documenting it at the same time. The developer of the code may defend it by explaining why it was made this way or asking any questions if she is unsure about the reviewer's comment. The third review meeting had a different flow since it was about reviewing tests. During the third review meeting both team members were performing as the developer and reviewer, discussing both opinions on the code quality and negotiating till reaching a mutual agreement.

2. Code review meeting 1:

	Developer	**Reviewer**
Name	Tatiana Sutulova	Elaf Abdullah Saleh Alhaddad
Student ID	30806151	31063977
Date and time	27/10/2020. 06:00pm -08:20pm	
Code reviewed	Methods of Calendar class: • **get_events_reminders_in_the_future(..)** • **get_event_by_keyword(..)** • **represent_events_reminders(..)**	
Summary of defects		

Method	Defect	Reasoning	Improvement
get_events_reminders_in_the_future()	General documentation of the code	While the method does not have high complexity, the method does not include explanation on how the information of the event is extracted from the API. The method does not include documentations regarding the parameters passed to the function	Include documentations that explain the method used to interact with the calendar API so that anyone reading the code will understand the algorithm of the function more. Include documentation explaining the arguments passed to the function.
	Line 65 (**years +=2**)	Hard coded value of the number of upcoming years that the function will extract the events from. This compromises the maintainability of the code, incase the requirements is changed and the user is able to get the upcoming events for variety of years (e.g 5 years instead of just 2)	The number of years can be passed as an argument.
	Parameter passed to the function, specifically **starting_time**	In line 64, the method generates the current time so that the calculation for the ending period is easier.	Eliminate the **starting_time** parameter and use current_time instead in line 65.

		This makes the parameter **starting_time** redundant.	
get_event_by_keyword(..)	The method has a large number of arguments	This causes two major problems: ● increases the dependency it has on the function called **main().** This function does not belong to the calendar class which could lead to privacy leak. ● This makes the code smell and indicates that the function is too big and complex.	Another method could be added that can be called from the function itself. This new method will be responsible for interacting with the user and getting the values returned to **get_event_by_keyword(..)** as an ArrayList instead of having them as arguments when it is called from the function **main().**
	Documentations and comments	The arguments passed to the function does not have any documentation that explains their part in the code. The arguments also have very vague naming conventions making it hard to understand	Add documentation on what these arguments stand for and their part in coding the functionality.

		what they actually stand for in the code (i.e **subtype**/**type**/**keyword**)	
	In-line comments	The code for this function does not have any in-line comments despite having a complex algorithm. This decreases the code readability	Add in-line comments that explain the algorithm of the code.
	Line 81 ~ line 93 and Line 101 ~ Line 112	This code is responsible for printing the event and the reminder of the event. However, the code in both lines are very similar and there is already a function that is responsible for printing the event and the reminder. Due to the duplication of this code, it comprises the overall code quality as the code is no longer DRY.	Use the function that already exists in the system **represent_events_reminders(..)** to print out the events to the user.
	Overall complexity of the method	The method handles too many responsibilities that does not represent the name of the function. The method name is to extract the searched event from the api.	One method called from inside this method that could interact with the user to get the event they are searching for. This method code can be changed so that it can extract the event from the Calendar api.

		However, the method seems to be passed the event as an argument and it will merely print it out to the user.	The **represent_events_reminders(..)** method can then be called to print out the event.
represent_events_reminders(..)	Line 229 (**defaultReminder**)	Does not follow naming convention	Change variable name to **default_reminder**
	Line 236~ Line 237	Variables defined (**default** and **override**) do not have long descriptive variable names that explains what do they stand for	Give these two variables longer names that describes what they stand for in the method accurately
	Documentation and comments	There are no comments that explain the parameters and the return value of the method	Add comments to explain the arguments passed to the method and the object returned by the method
	In-line comments	In-line comments are required for this functionality. The code algorithm is not very complex however, it is complex enough to require comments to increase the code readability.	Include in-line comments that explains the algorithm of the method
main(..)	line 295~306 (**subtype/type**)	Introduction of new variables that don't have descriptive names and are not understandable unless the reviewer has a greater understanding of the Calendar	Give the variables a more descriptive name that would increase the code readability.

		API.	
	Line 313	defaultReminder variable does not follow the naming convention	Change **defaultReminder** to **default_reminder**
	Documentation	The function has brief documentation on what is its responsibility. However, more documentation should have been added to explain how the function runs and implements this responsibility.	Change **"""Allows the user to interact with the calendar, works as an I/O console"""** to **"""Allows the user to interact with the calendar, works as an I/O console. Based on the input given by the user, the function will call the appropriate calendar methods and prints the return value to the user"""**
	Line 286 ~ line 318	The method seems to handle an extra responsibility other than just being a menu. This method seems to be interacting with the user and searching for the event instead of the method **get_event_by_keyword(..)**	The function should call **get_event_by_keyword(..)** and let that function handle getting the inputs and searching for the event. This function should only be responsible for displaying the menu and calling the methods based on the user choice from the menu.

3. Code review meeting 2:

	Developer	Reviewer
Name	Elaf Abdullah Saleh Alhaddad	Tatiana Sutulova
Student ID	31063977	30806151
Date and time	29/10/2020. 06:00pm -07:48pm	
Code reviewed	Methods of Calendar class: • get_past_events(..) • get_events_by_timeline(..) • delete_event_reminder(..)	

Summary of defects			
Method	**Defect**	**Reasoning**	**Improvement**
get_past_events(..)	Documentation and comments.	Comments do not explain what parameters were passed and they do not properly explain the functionality of the function.	Description of parameters that are passed to the function should be added. Moreover, it should describe what the function returns and should not describe what function is not actually doing.
	Inline comments	**# Implementing the functionality** This comment is redundant and does not give proper explanation	The redundant comments should be either replaced with more useful comments or removed.

	Hardcoded value **days=+1826**	Since the function is looking for events and reminders in the last 5 years, there is a hardcoded value related to it. If the function is to be used for another number of years, the value will have to be replaced	The number of years can be passed as a parameter and the hardcoded value will be replaced with number_of_years*365.
	Parameter passed to the function, specifically **starting_time**	In line 124, the method generates the current time so that the calculation for the ending period is easier. This makes the parameter **starting_time** redundant.	Eliminate the **starting_time** parameter and use current_time instead in line 65.
	Code does not handle all the external failures.	The function does not handle API failures	If API access is giving an error or accessing API is taking too long, the timeout error should be raised and the proper message should be represented to the user.
get_events_by_timeline(..)	Comment and documentation.	The complexity, exceptions raised and description of parameters passed is not included in the comment.	Describe the complexity, exceptions raised and what parameters are passed to the function
	Inline comments	There are not enough inline comments to explain the complex functionality of this method	Add more inline comments that will explain what all the if..else conditions are responsible for

	if year % 4 == 0 and (year % 100 != 0 or year % 400 == 0): ** date = int(input('Insert day date (from 1 ~ 29)'))** ** if date < 1 or date > 29:** ** raise ValueError("Date must be between 1 and 29 ")** **else:** ** date = int(input('Insert day date (from 1 ~ 28)'))** ** if date < 1 or date > 28:** ** raise ValueError("Date must be between 1 and 28 ")** Leap year calculation is met in the code two times.	There are several repetitive parts of the code that can be put outside of its scope(if..else condition) decreasing the repetition.	The code can be divided into smaller parts creating functions responsible for certain functionality for example calculating leap year. These functions will be called when needed.
delete_event_reminder	Comment and documentation.	The complexity, exceptions raised and description of parameters passed is not included in the comment.	Describe the complexity, exceptions raised and what parameters are passed to the function
	check = input("would you like to delete: \n 1. event \n 2. exit \n") **if check == "1":** **api.events().delete(calendarId='primary', eventId=eventId, sendUpdates='all').execute()** ** print("Event deleted successfully")** **elif check == '2':** ** print("Event is not deleted")** **else:**	There is a certain level of code duplication in this function, that may be reduced in order to make the code DRY.	This problem can be solved by writing these lines of code outside of the if.. else statement block.

	raise ValueError("Invalid Input") This part of code is repeated in different if and else scopes.		

4. Code review meeting 3:

Date and time	27/10/2020. 06:00pm - 08:20pm		
Code reviewed	Test cases and suites for the methods in CalendarTest.py		
Summary of defects			
Method - Testing	**Defect**	**Reasoning**	**Improvement**
get_past_events() **get_events_reminders_in_the_future()**	Comments and documentation	**# This is because the function is dependent on the api() and has no conditions or branches** The header comment is starting with this sentence, which makes the comment incomplete and reduces understandability. **# Test 1: Getting the future events from** This comment is in the function that actually tests the past events	The comment should be rephrased to make more sense and comments must be properly checked to make sure they properly explain what code is responsible for.

	Code duplication	There are a lot of lines of code that exist in both methods, making the code repetitive.	Both methods may be combined into one method, which will significantly reduce the repetition.
	Redundant code	The method that is responsible for testing **get_past_events()** has the same line of code repeated two times: **time_now = '2020-10-16T16:59:25.423Z'** Exists on line 118 and 122	Remove one of the lines
get_events_by_timeline()	Comments and documentation	There are several parts that are complex enough to require comments, however proper inline comments are missing	Include more comments that will explain purpose of complex variables and conditions
	Code duplication	The significant part of code in all the methods is repeated from one method to another, since all the methods are using the same approach with different hardcoded values.	All the methods may be combined into a single one, which will have two arrays with hardcoded inputs and expected outputs and a for loop that will iterate through them simultaneously. The loop will be testing the new possible combination every iteration.
get_events_by_keyword()	Line 508	Variable **defaultReminder** does not follow the proper naming convention	Change **defaultReminder** to **default_reminder**

	Line 519~521	Introduction of variables **subtype**, **type** and **key_word**. These two variables have very vague naming conventions making it hard to understand what they actually stand for in the code.	Change the arguments names to be more descriptive of what they stand for in testing the functionality.
	Line 519~520 and Line 527~528 Line 552 ~553 and Line 560 ~561	These couple of lines assign the variables subtype and type to the same value. These is unnecessary and is an indication of duplicated code	Assign the variables once and only reassign it again when you want to change the variable value
	Documentations	The documentations are not sufficient as they do not explain what are the parameters that were passed to the function	Add the parameters description in the documentation
	In-line comments: ● Line 544 (**# test 3: F, T, F ->F (raises an assertion)**) ● Line 558 (**# test 5: T, T, F**) ● Line 562, Line 567 and Line 572 (**# event with the following keyword does not exist**)	Line 554: does not specify or explain why the code should raise an assertion Line 558: does not explain what does the condition mean and what are the consequences of this condition	Line 554: Add an explanation on why would the code raise an assertion error Line 558: Add an explanation on what does the condition stand for and what is the expected consequences Line 562, 567, 572: Fix the comments so that they accurately represent what the

		Line 562, 567, 572: The comment says that the keyword does not exist in the api which means that the code should raise an error. However, the keyword exist in the api and that is why it prints the specified outcomes in Line 564, 569 and 574	line of code is actually doing and what is it testing
represent_events_reminders()	Naming conventions	Some naming conventions are not descriptive enough: **events_no** Does not explain properly what this variable is responsible for	Make variable names more descriptive
	Comments	**# Don't change the formatting of this one as it will raise an error due to it being in a different formation** **# of the expected string** This comment is repeated a lot of times, before every string.	The repeated comment could have been mentioned once in the header comment.
delete_event_reminder()	Repetition of the initialisation of mock_api in each method	This makes the code smell as it is no longer DRY due to these redundant duplication	Initialise the mock_api in the setUp constructor for these testing methods
Overall			

5. Conclusion

Reviewing the code allowed us to spot some mistakes that we didn't think of while coding the Calendar. It also helped us in understanding the different styles of our coding and how they can be improved to follow the Python programming rules and constrictions.

Overall, the code written does not have major faults that would compromise the application. Most of the code improvements are to ensure that the code follows coding requirements and reduce code duplication to keep the code DRY. However, we noticed major flaws in the documentation of the code. They were either not sufficient enough to explain complex functionalities or had repetitive comments that did not necessarily apply to the lines of code it was referring to.

6. Appendix

6.1 Code review meeting 1:

	Developer	**Reviewer**
Name	Tatiana Sutulova	Elaf Abdullah Saleh Alhaddad
Student ID	30806151	31063977
Date and time	27/10/2020. 06:00pm -08:20pm	
Code reviewed	Methods of Calendar class: • get_events_reminders_in_the_future(..) • get_event_by_keyword(..) • represent_events_reminders(..)	

6.1.1 Method get_events_reminders_in_the_future(...)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	All variables are descriptive and follow the python naming convention, making them easy to understand and self explanatory	**current_time** and **ending_time** equates to the time period to represent the events from.	-

				Where **current_time** is the current starting time and **ending_time** is the ending time.	
2	Is the scope for all the variables appropriate ?	Y	The variables used are defined inside the method and so the scope is local and within the method	-	-
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	Y	Method responsibility and algorithm is simple. No intensive documentation required.	-	Explanation for the line code that extracts the event from the API can be added to ensure that any programmer not familiar with the API can understand. Explanation for returned object and parameters can be included.
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	N	Method responsibility and algorithm is simple. No intensive documentation required.		

5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N			
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	Y	In case the functionality changes and we would like to show the event reminders for the upcoming 5 years instead of two. We would require to change the number from +=2 to +=5.	"**years+=2**" in line 65.	The number of years can be passed as a parameter
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	N	-	-	-
3	Are there any parts of the code that do not benefit the program or not used at all?	N	-	-	-
4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	Y	The parameter **starting_time** can be eliminated without affecting how the method functions, making this parameter redundant	-	Remove the parameter (**starting_time**) and use the variable (**current_time**) that was defined locally instead.

5	Are the functions small and handle only small responsibility?	Y	-	-	-
Speed and performance					
1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	-	-	-
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	-	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	Python inbuilt function to define date and time were used instead of defining date and time manually	-	-
2	Is there any possible way to implement certain functionality with the simpler	N	-	-	-

	approach, which is more understandable and has a better performance?				
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	N	Method is simple and does not interact directly with the user. This means that the coding for this method doesn't require to be failure-tolerant	-	-
2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?	N	Method is simple and does not interact directly with the user. This means that the coding for this method doesn't require any error messages to be displayed to the user.	-	-

6.1.2 Method get_event_by_keyword(...)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	N	Variables that were defined inside the function are pretty simple and do not require any detailed explanation.	Only **count** and **reminder** are defined locally. Most of the other	-

				variables used were parameters passed to the function.	
2	Is the scope for all the variables appropriate ?	N	Most of the variables that were used inside the function were passed as parameters to the function. This makes the function way more dependent on outer functionalities.		Figure out a way to get these variables locally or from inside the function.
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	N	The parameters don't have descriptive names and comments didn't include any explanation for the parameters making it harder to understand.	**subtype/type/keyword**	Add comments that explain what these parameters stand for.
4	Are there in-line comments that are explaining the complex processes that	N	There were no in-line comments used. Considering the functionality is	-	Include in-line comments to explain

	may be not obvious for people who did not take part in writing this part of the code?		complex and hard to understand, this is a really important factor that will affect the readability and understandability of the code.		the complicated algorithm of the function
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	N	-	-	-
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	Y	The way that the reminders were printed could be changed to decrease the complexity of the function.	-	Use the method (**represent_events_reminders(..)**) that is already defined in the code.
3	Are there any parts of the code that do not benefit the program or not used at all?	N	-	-	-
4	Are there too many parameters in the function? Can it be reduced by reducing	Y	There are too many parameters that are passed to the function which makes it heavily dependent on **main(),**	**api/common_events/ subtype/type/ key_word/**	Split the function in half one that would be responsible for getting

	dependencies/ making the function smaller?		a function that does not belong to the calendar class.	**defaultReminder**	the inputs of the user instead of main. Another one that will be used to get the event based on the inputted values of the user.
5	Are the functions small and handle only small responsibility?	N	Function handles representing the reminders even if there is already a function that handles that responsibility.	-	Use the method (**represent_events_reminders(..)**) that is already defined in the code.
Speed and performance					
1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	-	-	-
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	-	-	-
Complexity					

1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	N	The code can't be simplified by using the python inbuilt functions.	-	-
2	Is there any possible way to implement certain functionality with the simpler approach, which is more understandable and has a better performance?	Y	The function can be approached in other ways to decrease the complexity of how it is implemented.	-	By reusing **represent_events_reminders(..)** to print the event and its reminders. By creating another function to decrease the number of arguments and dependency on the main() function
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	N	Method does not interact directly with the user. This means that the coding for this method doesn't require to be failure-tolerant	-	-

2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?	N	Method does not interact directly with the user. This means that the coding for this method doesn't require any error messages to be displayed to the user.	-	-

6.1.3 Method represent_events_reminders(...)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	N	Some variables are not easy to understand and would require a longer, descriptive name. **defaultReminder** does not follow python naming convention.	**Override** and **defaultReminder**	Give **override** a more descriptive name and fix the naming convention of **defaultReminder** to **default_reminder**.
2	Is the scope for all the variables appropriate ?	Y	All variables used are defined locally.	-	-
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function,	N	There are no sufficient comments that explain the parameters and the return value of the functionality.	-	Add comments to explain parameter and the object return

	does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity				
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	N	In-line comments are required for this functionality. The code algorithm is not very complex however, it is complex enough to require comments to increase the code readability.	-	Include in-line comments that explains the algorithm of the method
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	N	-	-	-
2	Are there repetitive parts of the code that	N	There are similar parts inside the code but they do not reduce the code	-	-

	will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?		readability or make it more complex		
3	Are there any parts of the code that do not benefit the program or not used at all?	N			
4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	N	-	-	-
5	Are the functions small and handle only small responsibility?	Y	-	-	-
Speed and performance					
1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	-	-	-
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right	Y	-	-	-

	functionality?				
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	N	The code can't be simplified by using the python inbuilt functions.	-	-
2	Is there any possible way to implement certain functionality with the simpler	N	The implementation of this function is simple enough	-	-

	approach, which is more understandable and has a better performance?				
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	N	Method does not interact directly with the user. This means that the coding for this method doesn't require to be failure-tolerant	-	-
2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?	N	Method does not interact directly with the user. This means that the coding for this method doesn't require any error messages to be displayed to the user.	-	-

6.1.4 Function main(..)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	N	In line 295~306, there is introduction of new variables that don't have descriptive names and are not understandable unless the reviewer has a greater understanding of the	**subtype/type**	Give the variables a more descriptive name that would increase the code readability.

			Calendar API. defaultReminder (line 313) variable does not follow the naming convention		Change **defaultReminder** to **default_reminder**
2	Is the scope for all the variables appropriate ?	Y	All variables used are initialised locally		
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	Y	The function does not require any of these documentations as the function does not have any return value or parameters. However, more documentation should have been added to explain how the function runs.	**"""Allows the user to interact with the calendar, works as an I/O console"""**	**"""Allows the user to interact with the calendar, works as an I/O console. Based on the input given by the user, the function will call the appropriate calendar methods and prints the return value to the user”””**
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	Y	-	-	-

5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	N	-	-	-
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	N	-	-	-
3	Are there any parts of the code that do not benefit the program or not used at all?	N	-	-	-
4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	N	-	-	-

5	Are the functions small and handle only small responsibility?	N	The function handles a large responsibility that could be split up into two smaller functions. The function is currently responsible for: ● Displaying the menu ● Getting the input choice from the user and calling the function corresponding to it ● Getting the input keyword to search for an event and then calling **get_event_by_keyword(..)**	-	Split the function into two smaller functions Function #1: ● Displays the menu ● Gets the inputs choice from the user and calling the function corresponding to it Function #2: ● Gets called inside **get_event_by_keyword(..)** ● Gets the inputs based on the user choice ● Returns the keyword the user is searching for to **get_event_by_keyword(..)**
Speed and performance					

1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	-	-	-
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	-	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	No inbuilt functions provided by python could decrease the complexity of this function	-	-
2	Is there any possible way to implement certain functionality with the simpler approach, which is more understandable and has a better performance?	Y	The function is currently responsible for: ● Displaying the menu ● Getting the input choice from the user and calling the function corresponding to it ● Getting the input keyword to search for an event and then	-	Split the function into two smaller functions Function #1: ● Displays the menu ● Gets the inputs choice from the

			calling **get_event_by_keyword(..)** It can be split into two smaller functions. That way it will decrease its complexity and decrease the dependency **get_event_by_keyword(..)** has on it.		user and calling the function corresponding to it Function #2: ● Gets called inside **get_event_by_keyword(..)** ● Gets the inputs based on the user choice ● Returns the keyword the user is searching for to **get_event_by_keyword(..)**
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	Y	The code will print out a statement for wrong inputs and will allow the user to enter another option after displaying the error message	This is implemented using the **while** loop (line 288 ~319)	-
2	Are the error messages user-friendly,	Y	The error message will indicate directly that the user put in the wrong input/	Message displayed is: **Invalid input, choose**	-

	properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?		unavailable option	**the number between 1-5**	

6.2 Code review meeting 2:

	Developer	**Reviewer**
Name	Elaf Abdullah Saleh Alhaddad	Tatiana Sutulova
Student ID	31063977	30806151
Date and time	29/10/2020. 06:00pm -07:48pm	
Code reviewed	Methods of Calendar class: • get_past_events(..) • get_events_by_timeline(..) • delete_event_reminder(..)	

6.2.1 Method get_past_events(...)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures	Y	All the variables have long and descriptive names	**current_time** variable representing the time at the moment	-

	that the role of each variable is understandable?				
2	Is the scope for all the variables appropriate ?	Y	All the variables used in this function are local and parameters sent to the function.	-	-
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	N	Comment does not explain what parameters were passed and it does not properly explain the functionality of the function.	**""" ...Prints the start and name..."""** The function does not print anything.	Description of parameters that are passed to the function should be added. Moreover, it should describe what the function returns and should not describe what function is not actually doing.
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	Y	There are several comments that help to understand what the complex lines of code are for.	**# calculation for the past 5 years** Explains that the lines following lines are responsible for calculations	-
5	Are there redundant comments that confuses the reviewer more than helping to understand	Y	There are comments that do not give proper explanations and seem redundant.	**# Implementing the functionality** This comment is	The redundant comments should be either replaced with

	the code?			general and does not make the code clear.	more useful comments or removed.
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	Y	Since the function is looking for events and reminders in the last 5 years, there is a hardcoded value related to it. If the function is to be used for another number of years, the value will have to be replaced.	**days=+1826** 1826 is 5 years translated to days	The number of years can be passed as a parameter and the hardcoded value will be replaced with number_of_years*365.
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	N	There are no duplicated parts in this function	-	-
3	Are there any parts of the code that do not benefit the program or not used at all?	N	All the parts of this function implement to the final result	-	-
4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	N	There are only two parameters in the function.	-	-

5	Are the functions small and handle only small responsibility?	Y	This function is short (about 20 lines) enough to understand it.	-	-
Speed and performance					
1	Are there any cases when the API or other external libraries are called more times than needed, which results in increase of processing time? Can this be reduced?	Y	Getting the current time requires accessing external libraries, it is being called twice in both methods	**current_time = datetime.datetime.today()** This call is done twice.	It could be called once in **main()** function and passed as a parameter to both of them
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	The code returns what it is supposed to return and works properly.	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	There is no place where the inbuilt python functionality could be used.	-	-
2	Is there any possible way to implement certain functionality with the simpler approach, which is more understandable and has a better performance?	N	The way the function is implemented is simple and understandable	-	-

Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	N	The function does not handle API failures	-	If API access is giving an error or accessing API is taking too long, the timeout error should be raised and the proper message should be represented to the user.
2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?	-	There are no messages since there are no raised errors.	-	-

6.2.2 Method get_events_by_timeline(..)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	Variable names are descriptive and follow naming conventions.	**starting_time** **ending_time** All the variable names are in the underscore convention and describe what the variable is responsible for.	-

2	Is the scope for all the variables appropriate ?	Y	All the variables are within the proper scope: either the function scope or condition scope.	-	-
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	N	The comment clearly explains the responsibility of the following function. However, the complexity, exceptions raised and description of parameters passed is not included	-	Describe the complexity, exceptions raised and what parameters are passed to the function
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	N	There are few inline comments. However, it's not enough to explain the complex functionality of the following method	-	Add more inline comments that will explain what all the if..else conditions are responsible for
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-

Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	N	There are a lot of hardcoded values, however they all are responsible for dates, months,years, etc., so even if the code is changed those values should always remain the same	**-**	-
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	Y	There are several repetitive parts of the code that can be put outside of its scope(if..else condition) decreasing the repetition.	**if year % 4 == 0 and (year % 100 != 0 or year % 400 == 0):** ** date = int(input('Insert day date (from 1 ~ 29)'))** ** if date < 1 or date > 29:** ** raise ValueError("Date must be between 1 and 29 ")** **else:** ** date = int(input('Insert day date (from 1 ~ 28)'))** ** if date < 1 or date > 28:** ** raise ValueError("Date must be between 1 and 28 ")** Leap year calculation is met in the code two times.	The code can be divided into smaller parts creating functions responsible for certain functionality for example calculating leap year. These functions will be called when needed.
3	Are there any parts of the code that do not benefit the program or not used at all?	N	-	-	-

4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	N	There is only one parameter passed to this function.	-	-
5	Are the functions small and handle only small responsibility?	N	The function may be splitted into several small functions, which will increase the maintainability of the code.	-	-
Speed and performance					
1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	-	-	-
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	-	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the	Y	No code may be replaced with the python inbuilt functionality.	-	-

	code is made as simple as possible without reimplementing existing python functionality?				
2	Is there any possible way to implement certain functionality with the simpler approach, which is more understandable and has a better performance?	Y	As it was mentioned before, the number of if conditions may be reduced for better understandability and maintainability of the code, however it does not affect the performance since the time complexity of if statements is O(1).	**-**	**-**
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	Y	All the improper user inputs are handled by raising exceptions.	**raise ValueError("Date must be between 1 and 31 ")**	**-**
2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should do to resolve it (if not obvious)?		Messages for the errors are descriptive helping the user to understand what went wrong and what they are supposed to do instead.	**"Date must be between 1 and 29 "**	**-**

6.2.3 Method delete_event_reminder(..)

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	All the variables have long and descriptive names	-	**-**
2	Is the scope for all the variables appropriate ?	Y	All the variables used in this function are local and parameters sent to the function.	**-**	**-**
3	Are the comments for methods, classes, and functions descriptive enough to understand its purpose without looking at the code? For example, for the function, does it include description of: - parameters passed to the function, - the value that the function returns - functionality - exceptions raised (if any) - time complexity	N	Comment properly describes the functionality of this function. However, parameters passed to the function and exceptions being raised in it as well as its complexity are not explained	**"""Deletes an event based on the event ID"""**	Explaining parameters passed to the function, exceptions being raised in it and its complexity in the comment.

4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	N	Method responsibility and algorithm is simple. No intensive documentation required.	-	-
5	Are there redundant comments that confuses the reviewer more than helping	N	No redundant comments in the following function	-	-

	to understand the code?				
Maintainability					
1	Are there any hardcoded values, which will be difficult to replace if the code is to be changed?	N	This function does not contain any hardcoded values	**-**	-
2	Are there repetitive parts of the code that will lead to difficulties if these parts are to be improved/replaced? Can we reduce duplication to make the code DRY?	Y	There is a certain level of code duplication in this function, that may be reduced in order to make the code DRY.	**check = input("would you like to delete: \n 1. event \n 2. exit \n")** **if check == "1":** **api.events().delete(calendarId='primary', eventId=eventId, sendUpdates='all').execute()** **print("Event deleted successfully")** **elif check == '2':** **print("Event is not deleted")** **else:** **raise ValueError("Invalid Input")** This part of code is repeated in different if and else scopes.	This problem can be solved by writing these lines of code outside of the if.. else statement block.

3	Are there any parts of the code that do not benefit the program or not used at all?	N	All the parts of this function implement to the final result	-	-
4	Are there too many parameters in the function? Can it be reduced by reducing dependencies/ making the function smaller?	N	There are only two parameters in the function.	-	-
5	Is the function small and handles only small responsibility?	Y	By reducing repetition the function might be smaller, however, currently it is small enough to properly understand it and maintain if needed.	-	-
Speed and performance					
1	Are there any cases when the API is called more times than needed, which results in increase of processing time? Can this be reduced?	N	Api is called twice: to check the properties of the reminder and to delete the event/reminder. It cannot be reduced, since both calls are necessary to implement the functionality	**-**	**-**
2	Does the code do what it is supposed to do? For example, do all the choices in the menu correspond to the right functionality?	Y	The function performs proper functionality and does what it is expected to do.	**-**	**-**
Complexity					

1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	There is no re-implementation of existing python functionality, the proper inbuilt functions are used when it's necessary.	**print(...)** Prints out to the console the string added inside the brackets **input()** Prompts the user to input the value.	-
2	Is there any possible way to implement certain functionality with the simpler approach, which is more understandable and has a better performance?	N	As it was mentioned before, the number of if conditions may be reduced for better understandability and maintainability of the code, however it does not affect the performance since the time complexity of if statements is O(1).	-	-
Reliability					
1	Is the code failure-tolerant, meaning that it handles various failures not related to the actual code, such as wrong user input, API failures, requests for non-existing data, etc.?	Y	The app raises an error if the user input is wrong.	**raise ValueError("Invalid Input")**	-
2	Are the error messages user-friendly, properly explaining the user what the issue is and hinting what the user should	Y	The comment is descriptive enough for the user to understand what went wrong.	**"Invalid Input"**	-

	do to resolve it (if not obvious)?				

6.3 Code review meeting 3:

	Developer	**Reviewer**
Name	Elaf Abdullah Saleh Alhaddad	Tatiana Sutulova
Student ID	31063977	30806151
Date and time	31/10/2020 06pm-8pm	
Code reviewed	Test suits and cases	

6.3.1 Testing for get_events_reminders_in_the_future() and get_past_events()

The tests for this functionality were split into two methods: • **test_get_events_reminders_in_the_future(..)** • **test_get_past_events(..)**					
#	**Description**	**Y/N**	**Explanation**	**Example (optional)**	**Possible improvement**
General					
1	Does the test case run?	Y	-	-	-
2	Does the test case cover most of the important code? Branches and conditions?	Y	The two methods being tested are simple and small, which requires a line coverage testing, which was easily	-	-

			reached by the test.		
3	Does the test case increase the total test coverage?	Y	Both methods increase the total coverage	-	-
4	Is the mocking properly used?	Y	Mocking is properly used to mock the API	**mock_api = Mock()**	-
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	All the variables used in these test methods have descriptive names that follow the naming convention (underscore convention)	**time_now** Which represents the time at the moment	**-**
2	Is the scope for all the variables appropriate ?	N	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method.	**mock_api = Mock()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
3	Are the comments for the test method descriptive enough to understand its purpose without looking at the code? Does it include description of: - parameters passed to the function,	Y	Comments are very detailed, describing all the details about this test case	**-**	**-**

	- what is being tested and what approach is used for that				
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	Y		-	-
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	Y	Comments must be more properly checked since they include confusing parts.	**# This is because the function is dependent on the api() and has no conditions or branches** The comments is starting with this sentence **# Test 1: Getting the future events from** This comment is in the function that actually tests the past events	The comment should be rephrased to make more sense.
Maintainability					
1	Are there repetitive parts of the code? Can we reduce duplication to make the code DRY?	Y	There are a lot of lines that exist in both methods making the code repetitive.	-	Both methods may be combined into one method, which will significantly reduce the

					repetition.Some parts may be put into the constructor since they are repetitive for all the test methods.
2	Are there any parts of the code that do not benefit the testing or not used at all	Y	The method that is responsible for testing the past events has the same code repeated two times.	**time_now = '2020-10-16T16:59:25.423Z'** Exists on line 118 and 122	Remove one of the lines
Speed and performance					
1	Are there any cases when the API is called in the testing which results in increase of processing time?	N	The API is mocked and not being called anywhere	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	-	-	-
2	Is there any possible way to implement this test with the simpler approach, which	Y	As it was mentioned before both methods could be combined into one.	-	-

	is more understandable and has a better performance?				

6.3.2 Testing for delete_event_reminder

The tests for this functionality were split into three methods:
- **test_delete_event_reminder_custom_reminder(..)**
- **test_delete_event_reminder_default_reminder(..)**
- **test_delete_event_reminder_no_reminder(..)**

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
General					
1	Does the test case run?	**Y**	-	-	-
2	Does the test case cover most of the important code? Branches and conditions?	**Y**	The method represents a test suite and runs multiple test cases to test the outcome based on the possible input. This ensures that all conditions and branches were covered for this method.	-	-
3	Does the test case increase the total test coverage?	Y	-	-	-
4	Is the mocking properly used?	Y	Mocking is used to mock the API, the input of the user and the possible	-	-

			output that will be displayed to the user.		
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	All the variables used in these test methods have descriptive names that follow the naming convention (underscore convention)	**event_reminder/ event_id**	-
2	Is the scope for all the variables appropriate ?	N	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method to avoid repetition.	**mock_api = Mock()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
3	Are the comments for the test method descriptive enough to understand its purpose without looking at the code? Does it include description of: - parameters passed to the function, - what is being tested and what approach is used for that	Y	Sufficient documentations were included that explains the purpose of the testing method provided.	-	-

4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	Y	Sufficient in-line comments included that explains each test that is conducted under the method. The comments also include the conditions, branches and explanation possible outcomes	-	-
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-
Maintainability					
1	Are there repetitive parts of the code? Can we reduce duplication to make the code DRY?	Y	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method to avoid repetition. Objects that were initialised in each function had similar codes however there is no available way that can make that code DRY without increasing its complexity (event_reminder/event_no_reminder are similar objects with only small variation)	**mock_api = Mock()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
2	Are there any parts of the code that do not benefit the testing or not used at all	N	-	-	-

Speed and performance					
1	Are there any cases when the API is called in the testing which results in increase of processing time?	N	API was mocked and hence not called for this test suite	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	inbuilt mocking and patching allowed the programmer to test the functionalities without connection to external factors other than the code.	-	-
2	Is there any possible way to implement this test with the simpler approach, which is more understandable and has a better performance?	N	Generally, the tests are highly readable due to the descriptive comments and variable names.	-	-

6.3.3 Testing for get_events_by_timeline()

The tests for this functionality were split into eleven methods:

- **test_timeline_valid(..)**
- **test_timeline_invalid(..)**
- **test_timeline_31_day(..)**
- **test_timeline_feb_leap(..)**

- **test_timeline_feb_not_leap**
- **test_timeline_30_days(..)**
- **test_timeline_feb_leap_specific_day(..)**
- **test_timeline_feb__not_leap_specific_day(..)**
- **test_timeline_30day_specific_day(..)**
- **test_timeline_31day_specific_day(..)**
- **test_timeline_with_no_events(..)**

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
General					
1	Does the test case run?	Y	-	-	-
2	Does the test case cover most of the important code? Branches and conditions?	Y	The approach used to test the get_events_by_timeline() is branch coverage. All the methods reach the 100% branch coverage.	-	-
3	Does the test case increase the total test coverage?	Y	All the methods increase the total coverage	-	-
4	Is the mocking properly used?	Y	Mocking is properly used to mock the API, user input/output and the other method that is called inside **get_events_by_timeline().**	**@patch('Calendar.Calendar.get_event_by_keyword') @patch('builtins.print') @patch('builtins.input', side_effect=['2021', '2', '2', '1', 'Event 10.02.2021'])**	-
Readability and understandability					

1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	Y	All the variables used in these test methods have descriptive names that follow the naming convention (underscore convention)	-	-
2	Is the scope for all the variables appropriate ?	N	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method.	**mock_api = Mock()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
3	Are the comments for the test method descriptive enough to understand its purpose without looking at the code? Does it include description of: - parameters passed to the function, - what is being tested and what approach is used for that	Y	The comment clearly describes what branch is being tested	-	-

4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	N	There are several parts that are complex enough to require comments, however proper inline comments are missing	-	Include more comments that will explain purpose of complex variables
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	N	-	-	-
Maintainability					
1	Are there repetitive parts of the code? Can we reduce duplication to make the code DRY?	Y	The significant part of code in all the methods is repeated from one method to another, since all the methods are using the same approach with different hardcoded values.	-	All the methods may be combined into a single one, which will have two arrays with hardcoded inputs and expected outputs and a for loop that will iterate through them simultaneously. The loop will be testing the new possible combination every iteration.

2	Are there any parts of the code that do not benefit the testing or not used at all	N	-	-	-
Speed and performance					
1	Are there any cases when the API is called in the testing which results in increase of processing time?	N	The API is mocked and not being called anywhere	-	-
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	-	-	-
2	Is there any possible way to implement this test with the simpler approach, which is more understandable and has a better performance?	Y	As it was mentioned before, methods can be combined into a single method, which will improve the code readability and maintainability, but the time complexity of the test case will be increased, which is not the desired result.	-	-

6.3.4 Testing for get_events_by_keyword()

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
General					
1	Does the test case run?	Y	-	-	-
2	Does the test case cover most of the important code? Branches and conditions?	Y	The method represents a test suite and runs multiple test cases to test the outcome based on the possible input. This ensures that all conditions and branches were covered for this method.		
3	Does the test case increase the total test coverage?	Y	-	-	-
4	Is the mocking properly used?	Y	Mocking is used to mock the API, the input of the user and the possible output that will be displayed to the user.	-	-
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	N	Just like the respective function, there are variables that don't have descriptive names that indicate what the variable represents if the person reading the code is not familiar with the calendar API.	type/subtype	Give these variables longer descriptive names and change defaultReminder to be default_reminder

			defaultReminder does not follow the proper naming convention		
2	Is the scope for all the variables appropriate ?	N	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method.	**mock_api = MagicMock()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
3	Are the comments for the test method descriptive enough to understand its purpose without looking at the code? Does it include description of: - parameters passed to the function, - what is being tested and what approach is used for that	N	The documentation describes what the function is testing. However, it does not describe the parameters that are passed to the function.	**-**	**-**
4	Are there in-line comments that are explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?	Y	There are in-line documentations that explain the branches that are being tested. Some of these comments are hard to understand as it doesn't explain what are the consequences of these conditions and how they are important.	**# test 5: T, T, F**	Elaborate on what do the conditions mean and how is the expected outcome different from other branches and conditions

5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	Y	Some of the comments are wrong and do not apply to the line of code they are corresponding to.	**# event with the following keyword does not exist** However, the keyword does exist in the mocked api and the code runs without raising an error	Change the comments to make them accurately represent what the line of code is actually doing.
Maintainability					
1	Are there repetitive parts of the code? Can we reduce duplication to make the code DRY?	Y	There is some duplicated code that can be reduced. The **subtype/type** is defined more than once with the same value in some parts and different values in others.	**subtype = 'organizer'** **type = 'email'** Present in line (519~520) and (527~528)	Instead define it once and change the value only when required to do so
2	Are there any parts of the code that do not benefit the testing or not used at all	N	-	-	-
Speed and performance					
1	Are there any cases when the API is called in the testing which results in increase of processing time?	N	The API is mocked and not being called anywhere	**-**	**-**

Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	inbuilt mocking and patching allowed the programmer to test the functionalities without connection to external factors other than the code.	-	-
2	Is there any possible way to implement this test with the simpler approach, which is more understandable and has a better performance?	N	However, the comments for these tests are crucial in making the code easier to understand.		Improve the comments for this code as they are confusing and decrease the readability of the code

6.3.5 Testing for represent_events_reminders()

#	Description	Y/N	Explanation	Example (optional)	Possible improvement
General					
1	Does the test case run?	Y	-	-	-
2	Does the test case cover most of the important code? Branches and conditions?	Y	The following test approach is condition coverage that covers 4 different conditions by 100%	-	-
3	Does the test case increase the total test coverage?	Y	The method significantly increases the total coverage	-	-

4	Is the mocking properly used?	Y	Mocking is properly used to mock the output of the function	**@patch('builtins.print')**	-
Readability and understandability					
1	Are the names of variables descriptive and following naming conventions, which ensures that the role of each variable is understandable?	N	All the name conventions in the method are used properly, however, some names are not descriptive.	**events_no** Does not explain properly what this variable is responsible for	Make variable names more descriptive
2	Is the scope for all the variables appropriate ?	N	Certain parts could be initialised in the constructor of the class instead of instantiating them locally in every method.	**calendar = Calendar()** This part is being initialised in every method of the class	All the statements that are repeated from method to method could be initialised in the constructor instead of duplicating them
3	Are the comments for the test method descriptive enough to understand its purpose without looking at the code? Does it include description of: - parameters passed to the function, - what is being tested and what approach is used for that	Y	Header comment is very detailed, describing all the details about this test case	**-**	**-**
4	Are there in-line comments that are	Y	Inline comments are descriptive, helping to understand the test faster.	**-**	**-**

	explaining the complex processes that may be not obvious for people who did not take part in writing this part of the code?				
5	Are there redundant comments that confuses the reviewer more than helping to understand the code?	Y	There are several repetitive comments that could be mentioned once instead of mentioning it for every branch	**# Don't change the formatting of this one as it will raise an error due to it being in a different formation # of the expected string** This comment is repeated before every string.	The repeated comment could have been mentioned once in the header comment.
Maintainability					
1	Are there repetitive parts of the code? Can we reduce duplication to make the code DRY?	N	There is no duplication in the following testing method.	-	-
2	Are there any parts of the code that do not benefit the testing or not used at all	N	-	-	-
Speed and performance					
1	Are there any cases when the API is called in the testing which results in increase of	N	The API is not called in the following test method	**-**	**-**

	processing time?				
Complexity					
1	Are the proper python inbuilt functions used in the code, which ensures that the code is made as simple as possible without reimplementing existing python functionality?	Y	-	-	-
2	Is there any possible way to implement this test with the simpler approach, which is more understandable and has a better performance?	N	-	-	-