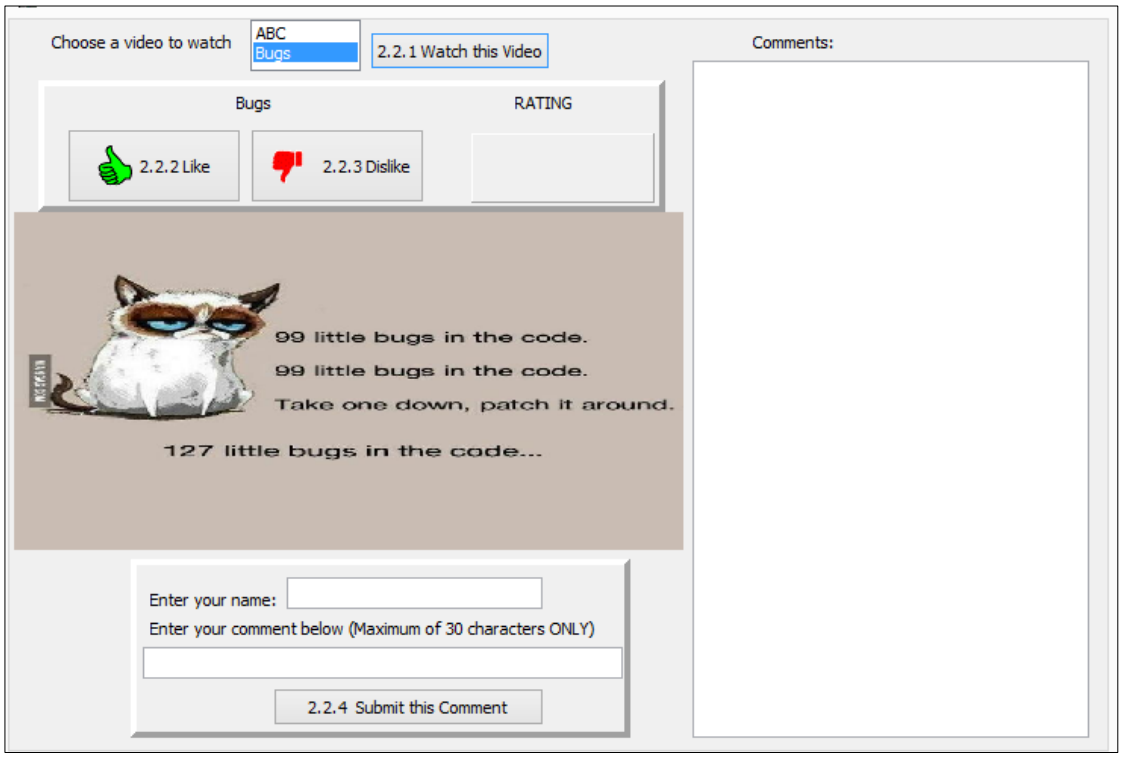
QUESTION 2: OBJECT-ORIENTED PROGRAMMING

Your school's computer club has started to develop a program where learners can comment, like or dislike videos that are available for viewing. Each video must store the amount of likes and dislikes and all comments associated with that video. Each video must also be rated according to the number of likes and dislikes associated with it.

Do the following:

- Open the incomplete program in the Question 2 folder.
- Open the incomplete object class **VideoClass.pas**.
- Enter your name as a comment in both **VideoClass.pas** and **Question2_u.pas**.
- Compile and execute the program. Currently the program has no functionality.
- Code has been written to load a picture file onto the image component provided and to display the panels provided. Do NOT remove or change any provided code.

The following user interface is displayed when the video, Bugs, has been chosen.



Complete the code for this program, as specified in QUESTION 2.1 and QUESTION 2.2.

2.1 The incomplete class **(TVideo)** contains the declaration of five attributes that describe the **objVideo** object.

NAMES OF ATTRIBUTES	DESCRIPTION
fvideoname	The name of the video
fcomments	A string containing a list of all comments including the name of the video, names of the learners who made the comments and the date of each comment
flikes	The number of likes for the video
fdislikes	The number of dislikes for the video
frating	The number of stars that the video is rated at

The following complete methods have been provided in the unit VideoClass.pas:

addLike, addDislike, toString

Complete the code in the object class, as described in QUESTION 2.1.1 to QUESTION 2.1.4 below.

2.1.1 Write code for a **constructor** method named **Create** that will receive the name of a video as a parameter.
Assign this parameter value to the correct attribute and initialise the attributes for the number of likes and dislikes to zero. (4)

2.1.2 Write code for a method named **AddNewComment** that will receive two string parameters which will each represent a comment and a date.
Set the value of the attribute **fcomments** as described below:

The two parameters must be joined to the attribute **fcomments** :

- An empty line must be added before the new comment is joined to the attribute.
- Add the new comment from the parameter list after the empty line is added.
- Add a tab space and then add the date from the parameter list to the string.
- An empty line must then be added after the date. (7)

2.1.3 Write a method named **SetRating** which will set a value to the attribute, **frating**, according to the following criteria:

- Subtract the number of dislikes from the number of likes.
- If the difference is less than or equal to 0, then the rating must be 0.
- The rating must be set to 5 for any value greater than 3, otherwise the rating must be set to 2. (7)

2.1.4 Write code for a method named **GetRating** which must return a string consisting of stars (*).
The string must contain stars (*) joined together depending on the value of the attribute, **frating.** Example: If frating is 2 then the string will consist of 2 stars. (**) (5)

2.2 An incomplete unit **Question2_u.pas** has been provided.

It contains code for the object class to be accessible and has an object variable **objVideo** already declared. It also contains a variable to hold the system date.

NOTE: The system date is provided and saved in a string variable named **SystemDate** in the unit **Question2_u.pas**

Global variables supplied: ***objVideo: tvideo; SystemDate: string;***

Code to calculate the system date and to display the picture file chosen from the list box has been provided. Do NOT delete or change any provided code.

The user will choose a video to view and code is provided in the onclick event of the list box named **lbxVideos** to load a picture onto the image component and show the panels for adding likes, dislikes and comments.

Follow the instructions below to code the solution:

2.2.1 **OnClick event of the list box named lbxVideos (TQuestion2.lbxVideosClick)**

The user chooses a video name from the list box, **lbxVideos**.
Write code to instantiate the object **objVideo** using the video name chosen**.** (3)

2.2.2 **Button bitbtnLike [2.2.2 Like] and Button bitbtnDislike [2.2.2 Dislike]**

In the onclick event of **bitbtnLike**, use the method of the class named ***addLike*** to add a 'like' to the video object

In the onclick event of **bitbtnDislik**, use the method of the class named ***addDislike*** to add a 'dislike' to the video object. (2)

2.2.3 **Button Q2_2_3btnView [2.2.3 Watch this video]**

Write code using a dialog box to allow the user to enter "Y" or "N" to the question: "Do you like this video?".

If the answer is "Y", then write code to execute the onclick event of the bitmap button named **bitbtnLike**, otherwise write code to execute the onclick event of the bitmap button named **bitbtnDislike.**

Add code using the methods of the class named **SetRating** and **GetRating** to update the rating and then display the updated rating in the panel named **pnlRatings**. (6)

2.2.4 Button Q2_2_4btnSubmit [2.2.4 Submit this comment]

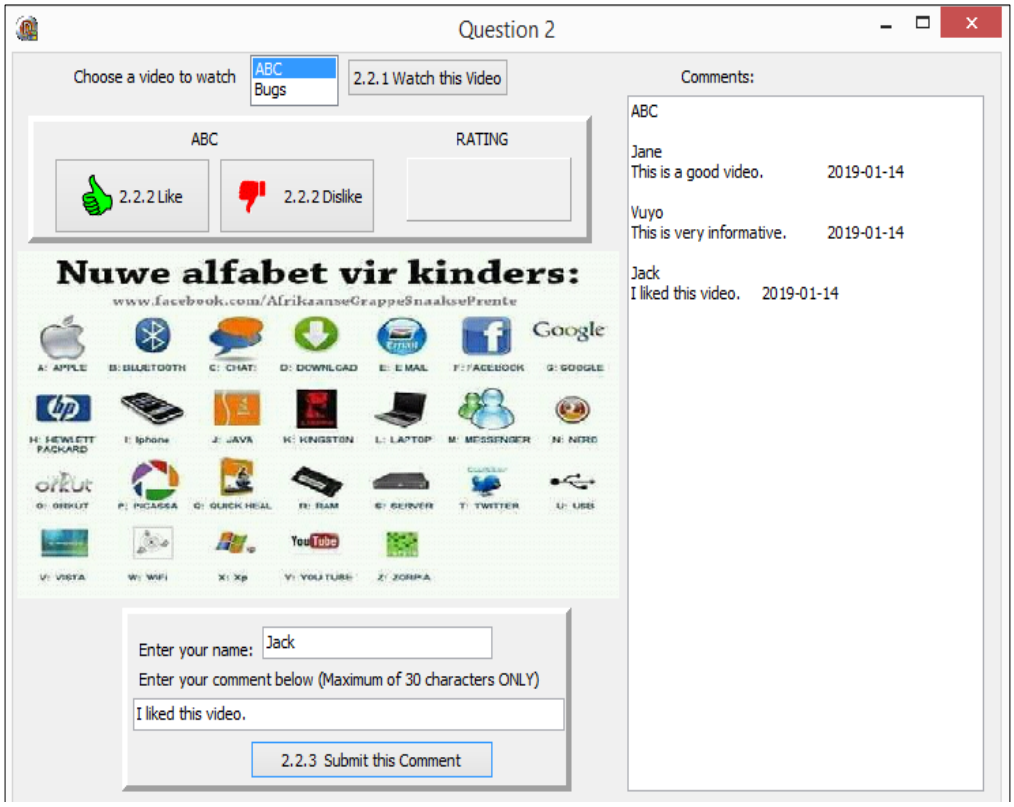
The user will enter a name and a comment in the edit boxes provided.

You must write code to join these two strings together with the comment on a separate line.
Example of compiled string: Jane Doe
This is a good video

Clear the richedit component named **redComments.**

Use the methods of the class named ***addNewComment*** (with the compiled string and system date as parameters) and ***toString,*** to update and display the comments in the richedit component named **redComments**.

Example of output for QUESTION 2.2.4:



(7)

- Enter your name and surname as a comment in the first line of the program file. (In both the class and the main program that uses the class)
- Save your programs.
- A printout of the code of both units may be required.

[41]