



Justin Cheung

Mr. Laserson

December 20, 2007

Introduction:

As a summative project, the TEE201 Computer Engineering class was told to create a program that would incorporate all the concepts learned during the Programming unit for the Turing Environment. All these concepts would be incorporated into a quiz program, which would test the user on a specific theme or topic. The entire project would be worth 10% of the final mark for the computer engineering course at Richmond Green Secondary School. Meanwhile, the following report will describe the quiz program done by Justin Cheung, the programming structures that were used to make the quiz program, and the problems that were encountered in the programming phase.

About the Science Quiz Program

Primarily, the program was created to quiz users on Grade 10 Science, mostly in the aspects of Chemistry and Ecology. The program consists of 15 questions; 5 multiple choice questions, 5 true or false questions, and 5 fill in the blanks questions. For each question, a different music plays in the background. If the user answers the questions correctly, an applause sound clip plays. Also, the counter will add 1 point to the final score. However, if the user answers incorrectly, a sound clip of someone screaming will play instead. Also, the correct answer to the question will be displayed. Once the user completes all the questions, the program will then display the mark that he or she received on the quiz and will be given an appropriate grade level according to the mark that they received.

Walk Through of the Science Quiz Program:

The Science Quiz program must be executed from the Turing Environment. This means that Turing must be installed on the computer in order to do the quiz. When the user first executes the program, a separate window will pop up, displaying a splash screen with the Science Quiz Program logo. After a delay of 5 seconds, the window will clear the screen and display a title page for the Science Quiz program. In order to continue the program, the user must type “start” and press the enter key on their keyboard. Afterwards, the user will be prompted to enter their name for the science quiz program. After the user enters his or her name, the enter key must be pressed to continue. Next, the window will display a menu with 4 different destinations: Quiz Rules, About Science Quiz, Start the Quiz, and Science Quiz Answers.

In the menu, the user will be prompted to enter the number that corresponds to their choice of destination. If they chose 1 as their destination, the window would clear its screen and then display the rules and regulations for the quiz. The rules will be displayed in two pages, consisting of the first three rules on the first page, and the next three rules on the second page – there are a total of six rules and regulations. To switch to the second page, the user must type “yes” and press the enter key. To exit the Rules and Regulations pages, the user must type anything of their choice and press the enter key. Then, the user would be prompted to enter their next destination. Typing anything and pressing enter would lead the user back to the menu. Typing “quiz” and pressing enter will prompt the quiz program to start the Science Quiz. In the last page of the Rules and Regulations section, the user will be prompted to choose their next destination automatically. If the user chose 2 as their destination, the program window would clear its screen and display general information about the Science Quiz Program and why it was created. To exit this section, the user will be prompted to type in their next destination. Typing “quiz” and pressing enter will prompt the program to start the quiz. Meanwhile, typing anything and pressing enter will lead the user back to the menu. If the user chose 3 as their destination, the program will initiate the Science Quiz. Lastly, if the user chose 4 as their destination, the program window will clear the screen and display the answers to the Science Quiz program in three pages. The first page displays all the answers for the Multiple Choice Questions. The second page displays the answers for the True or False Questions, and the third page displays the answers to the Fill in the Blanks questions. The answers are displayed in a similar manner as the Rules and Regulations section. The user must type “yes” and press the enter key to switch pages. If the user types anything else, they will be prompted to choose their next destination. Typing “quiz” and pressing enter will initiate the Science Quiz. Meanwhile, typing anything else would send the user back to the menu page.

Once the user initiates the Science Quiz from the Science Quiz program, the program menu will immediately clear the screen and display a title page for the Multiple Choice Questions. The title page will automatically clear away after a delay of 5 seconds. Afterwards, the user has to navigate through the first five questions, all multiple choice questions. For each multiple choice question, the program window will usually display the question and the choices for the first 4 seconds. Then, the window will clear the screen and display each choice for the question separately with images that correspond to those choices. There are 5 choices for each multiple choice question and each choice is displayed one after another after a delay of 4 seconds for each choice. The program window switches to the next choice by clearing the screen. After displaying all 5 choices, the user will be brought back to the question page where they will be prompted to enter the number that the answer corresponds to. Upon answering the question, the program window will display a message stating whether the answer was correct or not. Then, the program window will clear the screen and go to the next multiple choice question. After the user has completed the multiple choice questions, another title page will appear, similar to the one before the multiple choice questions. This time, however, the title page is for the True or False Questions.

Next, the user must navigate through questions 5 to 10, the True or False Questions. For each True or False question, the window will display a statement about Science and will prompt the user to enter in “t” for true, or “f” for false. Pressing enter will tell the program your answer. If the answer happens to be correct, the program window will clear its screen and display a message stating that the answer was correct along with a image that corresponds to the answer. Otherwise, if the answer happens to be incorrect, the program window will display a message telling the user that his or her answer was wrong. After the user completes the True and False Question, the last title page will appear for the Fill in the Blanks Question.

Lastly, the user must navigate through questions 11 to 15, the Fill in the Blanks questions. For each Fill in the Blanks question, the window will display the question and the blanks that have to be filled. It will also display a prompt, asking if the user if he or she wants a hint to the problem or not. If the user types in “yes” and presses the enter key, the program window will clear the screen and display a hint to the question along with an image that corresponds to the hint. Then, after a 10 second delay, the program window will clear its screen once more and display the question again. This time, the user is prompted to enter the answer to the Fill in the Blanks question. If the user chooses not to receive a hint, the program will skip the hint and immediately prompt the user for the answer. Upon entering the answer to the Fill in the Blanks question, the program will determine if the answer was correct or not. If the answer was correct, the program window will clear the screen and display a message saying that the answer was correct, along with an image that corresponds to the answer. If the answer was incorrect, then the program window will clear the screen and display a message stating that your answer was incorrect. Once the user has finished answering all the Fill in the Blanks questions, the program window will clear the screen and display a large image, saying, “Congratulations for Completing the Science Quiz.” The user is then congratulated at the same time with a sound clip of a large applause.

After displaying the congratulations image, the program window will clear the screen and display your mark for the Science Quiz over 15. It will then give the user a percentage grade and a letter grade, which corresponds to the percentage grade that the user received. The program window will also display a certain message, depending on the grade they got. For example, if the user got 60% to 70%, he or she would be told to work harder. For users whom achieved a perfect score on the quiz, the program will ask the users whether they want a certificate of achievement or not. If the user wants the certificate, they must type in “yes” and press enter. Afterwards, the program window will clear the screen and display instructions for receiving the certificate, telling the user to pause the program once the certificate shows up on screen. Then, after a delay of 10 seconds, the program window will display a certificate with the name of the user on it. This certificate can be printed out by pressing the print button on the window. Then, the user will be prompted to type “yes” to continue the program. Meanwhile, for users whom had not received 100%, the program will prompt the user to type “yes” to continue the program. If they type anything else, the program will redisplay your marks for the quiz. Once the user types “yes” and presses the enter key, the program will clear the screen and display another section which asks the user if they want to redo the quiz or to end it. Typing “end” and pressing the enter key will terminate the program. However, if the user types anything else, the program will loop back to the beginning of the program.

Programming Structure of the Science Quiz Menu:

The following will describe the programming structures used during the beginning of the program and in the Science Quiz Menu.

A loop was implemented from the beginning of the program to the end of the program. The purpose of this loop is to allow users to redo the quiz if they wished. At the end of the quiz, the user would be prompted to either redo the quiz or end it. The “**exit when**”structure, a programming structure for exiting loops, was used in this case, where “**r**”would be the variable for the prompt and the condition would be if r=”end”. In this case, the variable “**r**”was declared as a string, because it consists of letters. If the user were to type “end” and press the enter key, the Science quiz program would immediately end the program, as “end” was the condition for ending the Science Quiz Program’s loop. Meanwhile, typing anything else would just send the user back to the beginning of the program.

After declaring the loop, the next few lines of code tells the program to play the Super Smash Bros. Brawl music and to display a splash screen, followed by a title page. The music plays for most of the beginning of the program until the user starts the quiz. In order to play the music, the “**Music.PlayFileLoop”** structure was implemented. “**Music.PlayFileLoop”** causes the program to continue playing the designated music until the program stops the music using the **Music.PlayFileStop** programming structure. Apparently, “**Music.PlayFileLoop”** is only capable of playing MP3 music files and WAV music files. The “**Music.PlayFileStop” structure** is implemented just before the start of the Multiple Choice questions to stop the Super Smash Bros. Brawl music from playing. On the actual program, the structure for playing the Super Smash Bros. Brawl music file looks something like this: **Music.PlayFileLoop (“SMBBrawl.wav”)**. Most of the music clips played on this program use the same structure with different music clip files. Meanwhile, for the splash screen, an image is displayed by the program for 5 seconds until the program clears the screen to display the title page. The structure for displaying the image looks something like this: **var title:int**, followed by **title:=Pic.FileNew("title.jpg")**, followed by **Pic.Draw (title,60,150,picCopy)**. In this case, the **“var**”was used to declare the variable for the image, which was “**title**”. The variable was declared as an integer. Then, the following structure, “**title:=Pic.FileNew("title.jpg")”** , was used to declare the image file that would be used, which in this case, is “**title.jpg**”.Next, the “**Pic.Draw**” structure was used to display the image on the actual program, at the coordinates (60,150) (this coordinate is the lower left corner of the image) on the program window. Following the display of the image, the structures “**delay (5000)**” followed by “**cls**”(programming structure for clear screen), are used to clear the screen after 5 seconds to display the title page. For the most part, all images on this program are displayed using the same structures, but different variables. This excludes the “**cls**” and “**delay**” structures.

Next, the program displays the title page. Firstly, another infinite loop was implemented so that the user would return to the same title screen if they did not type “start” and press the enter key. Secondly, the structure “**drawfillbox(0,0,700,500,black)**”is used to display a black filled box from lower left corner coordinate (0,0) to top right corner coordinate (700,500). This black filled box is used just to fill in remainder of the program window not occupied by the picture that is displayed after. Thirdly, a large image of a frog with the logo for the Science Quiz program is displayed using the same image displaying structure as the one used for the splash screen. In this case, the image file is named “**titleGT.jpg**”, the variable for displaying the image is title1, which was displayed as an integer, and the coordinates in which the picture is placed on the program window is (0,0). Fourthly, the variable “**quit1**” is declared as a string as it contains words. Fifthly, the structure “**locatexy (75,100)**” is used to display the following prompt message in that specific coordinate. Sixthly, the prompt message is displayed using “**put”**, followed by the displayed message in quotations. All messages and questions displayed in words use “**put**” to output this information to the program window. Seventhly, the “**locatexy**” structure is used again to display what the user inputs at the coordinate (75,50). Eighthly, “**get**”, followed by the variable “**quit1**” allows the user to input the word “start”. Ninthly, the “**exit when**” structure is implemented so that when the user types “start” and presses enter, the title page loop is terminated. Tenthly, the “**end loop**” structure is used to end the title page loop. Lastly, the “**cls”** structure is used to clear the screen to display the prompt for inputting the user’s name.

After the title page, the user is prompted to enter their name. Firstly, the prompt image telling the user to enter their name is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. For this image, the variable “**name**”, declared as an integer, is used. The file name for this image is “**nameGT.jpg**”. Secondly, the variable “**name1**” is declared as a string. This variable will be used to store the user’s name. Thirdly, the “**locatexy**” structure is used to display the name that the user inputs at the coordinate (70,250). Fourthly, “**get**”, followed by the variable “**name1:\***” is put in to allow the user to input their name. The colon and the star beside “**name1**” will allow the user to enter a name with more than one space. Lastly, the “**cls**” structure is used to clear the screen and display the Science Quiz Program menu.

After the user has inputted their name, the program will bring them towards the Science Quiz Program Menu. Firstly, at the start of the Science Quiz menu, a loop is implemented so that the user can return to the menu from another sub category in the menu except for “**3. Start the Science Quiz**”. Secondly, the variables “**menu**” and “**dest1**” are declared as strings. The variable “**menu**” allows the user to choose their destination out of the 4 available ones. Meanwhile, the variable “**dest1**” allows the user to either return to the Science Quiz Menu from one of its destinations, or to start the Science Quiz. Thirdly, the menu options are displayed using “**put**”, followed by the options in brackets. Spaces in between the options are displayed using the following structure “**put “”**”, which causes the program to leave a blank space wherever this structure is displayed. Fourthly, an image that displays the title for the Science Quiz menu is displayed using the “**PIc.FileNew**” and “**Pic.Draw**” structure. Fifthly, the decision structure is created, where “**menu**” can equal 1, 2, 3, or 4 (These numbers correspond to the destination). If the user happens to enter 3, the “**exit when**” structure will clear the screen using “**cls**” and will terminate the menu loop and to start the science quiz.

If the variable “**menu**” equals 1, the program window will clear the screen using “**cls**”. Then following programming structure is then used to display the rules and regulations for the Science Quiz. Firstly, the variable “**page**” is declared as a string using “**var**”. This variable will allow the user to switch pages in the rules and regulations section. Secondly, eight blank spaces are displayed using “**put “”**”. Thirdly, the program window will display the image of the title for the Rules and Regulations section using the “**Pic.FileNew**” and “**PIc.Draw**” structures. Fourthly, the first three rules for the Science Quiz are displayed using “**put**”. A blank space is put between each rule using “**put””**”. Fifthly, a prompt message is displayed using “**put**”, asking if the user would like to switch to the second page. Sixthly, the prompt message is answered using “**get**”, followed by the variable “**page**”, which will allow the user to input whether they want to change pages or not. Seventhly, a decision structure is implemented, where if “**page**” equals “yes”, then the next three rules will be displayed in a similar manner to the first three rules, with the same spacing and image. However, if the inputted phrase is not “yes”, the program will display a prompt using “**put**”, asking if the user if they want to start the quiz or return to the menu. Then, the prompt message is answered using “**get**”, followed by the variable “**dest1**”, which was declared earlier. Following that, the “**exit when**” structure is used to terminate the Science Quiz Menu loop and to start the Science Quiz when the user inputs “quiz” and presses the enter key. However, if the user enters something different, the program will bring the user back to the menu. The same prompt message that asks the user if they want to return to the menu or start the quiz is also existent in the second page of the Rules and Regulations section for the Science Quiz. Eighthly, “**end if**” is used to end the decision structure for the variable “**page**”. Ninthly, “**else**” is used to declare what happens if the user does not choose to the start the quiz. Tenthly, “**cls**” is used to clear the screen of the program window, “**exit when**” tells the program to terminate the Science Quiz menu loop, and “**end if**” is used once again to end the decision structure for the Rules and Regulations section.

If the variable “**menu**” equals 2, the program window will clear the screen using “**cls**”. Then following programming structure is used to display information about the Science Quiz program and why it was created. Firstly, it displays a paragraph of information using “**put**”, along with blank spaces before and after the paragraph using “**put “”**”. Thirdly, an image of the title for the “About the Science Quiz” section is displayed using the structures “**Pic.FileNew**” and “**Pic.Draw**”. Fourthly, the variable “**dest2**” is declared as a string using “**var**” – this variable will allow the user to choose if they want to return to the menu or start the Science Quiz. Fifthly, a prompt message is displayed using “**put**”, asking the user if they want to start the quiz or return to the menu. Sixthly, “**get**”, followed by the variable “**dest2**”, allows the user to input their destination in relation to the prompt message. Seventhly, “**cls**” is used to clear the screen of the program window. Eighthly, the “**exit when**” structure is used to terminate the Quiz Menu loop when the variable “**dest2**” equals “quiz”. Ninthly, “**else**” is used to declare what happens if the user does not choose 2 for the variable “**menu”**, in which they will be sent back to the Science Quiz menu to choose another destination. If this is the case, the program window will clear the screen using “**cls**”. Then, if the user’s next destination is 3, the “**exit when**” structure will terminate the Science Quiz Menu loop and start the Science Quiz. Lastly, the “**end if**” structure is used to end the decision structure for the “About the Science Quiz” section.

If the variable “menu” equals 4, then the program window will clear the screen using “**cls**”. Then following programming structure is used to display the answers to the Science Quiz questions. Firstly, the variable “**page**” is declared as a string using “**var**”. This variable will allow the user to choose whether they want to change the page or not. Secondly, eight blank spaces are displayed using “**put “”**”. Thirdly, and image that displays the title for the Answers section is displayed using the structures “**Pic.FileNew**” and “**Pic.Draw**”. Fourthly, the answers to the 5 multiple choice questions are displayed using “**put**”. Fifthly, a prompt message is displayed asking the user if they want to change the page. This is followed by “**get page**”, which will allow the user to input their decision on whether or not they want to change the page. Sixthly, a decision structure is displayed, where if the variable “**page**” equals “yes”, then the program will clear the screen using “**cls**” and display the answers to the 5 True or False Questions in a similar manner to the answers for the 5 Multiple Choice Questions. Seventhly, the variable “**page5**” is declared as a string – this variable will allow the user to choose whether they want to switch to the next page or not. Eighthly, another prompt message is displayed using “**put**”, asking if the user wants to change pages. Ninthly, another decision structure is implemented, where if “**page5**” equals “yes”, then the program window will clear the screen using “**cls**” and display the answers to the 5 Fill in the Blanks Questions in a similar manner to the answers displayed for the 5 Multiple Choice questions and the 5 True or False questions. However, the image that displays the title for the answers will disappear after five seconds using the “**delay**” and “**cls**” structures – this is done to make room for the answers to the Fill in the Blanks Questions. Tenthly, the program window will display a prompt message using “**put**”, asking whether the user wants to start the quiz or return to the menu. Then, the “**get** dest1” allows the user to input their choice of whether to start the Science Quiz or to return to the menu. Following that, the “**exit when**” structure is used to terminate the Science Quiz menu program when “**dest1**” (A variable declared at the start of the Science Quiz Menu) equals quiz. Next, “**else**” is declared two times; once to declare what happens when the user chooses not to flip to the second page and again when the user chooses not to flip to the third page. For each of these “**else**” structures, the program window will display a prompt message using “**put**”, asking whether the user wants to start the quiz or return to the menu. Then, the “**get** dest1” allows the user to input their choice of whether to start the Science Quiz or to return to the menu. Following that, the “**exit when**” structure is used to terminate the Science Quiz menu program when “**dest1**” equals quiz. Afterwards, the “**end if**” structure is used to end each of the decision structures for the variables “**page**” and “**page5**” (The variables used to change the pages). Then, another “**else**” structure declares what happens if the user does not choose 4 as their destination for the variable “**menu**”, in which the program window will clear the screen using “**cls**”. Next, the “**exit when**” structure is used to terminate the Science Quiz Menu and start the Science quiz when “**menu**” is equal to 3. Lastly, the Science Quiz answers is ended with “**end if**”, which ends the decision structure for the variable “**menu**, and “**end loop**”, which ends the loop for the entire Science Quiz Menu.

Programming Structure for the Title Screens:

Before each set of questions, a title screen is displayed, telling the user whether they are about to answer Multiple Choice Questions, True or False Questions, or Fill in the Blanks Questions. Each title screen is actually a large image that is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Before the image is displayed, “**drawfillbox(0,0,700,500,black)**”is used to display a black filled box from lower left corner coordinate (0,0) to top right corner coordinate (700,500). This black filled box is used to fill in remainder of the program window not occupied by the image that is displayed afterwards. After the title screen is displayed, there is a delay for 5 seconds using “**delay (5000)**”. Then, the program window clears the screen using “**cls**” and starts displaying the questions for the Science Quiz.

Programming Structure for the Science Quiz:

The following will describe the programming structure used in the Multiple Choice Questions, the True or False Questions, and the Fill in the Blanks Questions.

For each of the Multiple Choice Questions, the following programming structure is used. Firstly, the variables for the answers are declared as strings using “**var**”. Secondly, a music file is played using the “**Music.PlayFileLoop**” structure. Thirdly, an image displaying all the choices is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Fourthly, the question is displayed using “**put**”. The question is displayed with a blank space using “**put “”**”before and after the question. Fifthly, “**delay (4000)**” and “**cls**” are used to display the question for 4 seconds before the program window clears the screen to display the choices individually. Sixthly, each of the five choices is displayed separately with images that correspond to the choice. The images for each choice are displayed using the “**PIc.FileNew**” and the “**Pic.Draw**” structures. The choices are displayed using “**put**” and the space above the choice is displayed using “**put “”**”. Then, for each of the choices, “**delay (4000)**” is used to display each choice for 4 seconds before “**cls**” is used to clear the screen to display the next choice. Seventhly, once all the choices have been displayed, the program window will clear the screen and display the question again, followed by the choices, using “**put**”. Eighthly, an image of all the choices is displayed again using “**Pic.FileNew**” and the “**Pic.Draw**” structures. Ninthly, prompt message is displayed using “**put**”, asking the user to input the letter that corresponds to the correct answer. Following that, “**get**”, followed by the variable for the answer allows the user to input the answer to the Multiple Choice Question. Tenthly, a decision structure decides on the correct answer for the answer variable. If the condition for the correct answer is true, the program window will display a message stating that the answer was correct using “put”. Then, the “**counter=counter+1**” will add 1 point to the counter, which counts how many correct answers that the user has. Next, the program will play an applause music file using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**” Otherwise, if the answer is incorrect, the program window will display a message stating that the answer was incorrect and will display the correct answer using “put”. Then, the program will play a music file with the sound of someone yelling, using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**” Lastly, the decision structure is ended using “**end if**”, delayed for 4 seconds using “**delay (4000)**”, and the screen is cleared using “**cls**”.

For each of the True or False Questions, the following programming structure is used. Firstly, the variables for the answers are declared as strings using “**var**”. Secondly, a music file is played using the “**Music.PlayFileLoop**” structure. Thirdly, eight blank spaces are displayed using “**put””**”. Then, the True or False Question is displayed using “**put**”. Fourthly, an image that displays “True or False?” is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Fifthly, four blanks spaces are displayed using “**put””**”. Then, the prompt message for inputting the answer is displayed using “**put**”. Afterwards, “**get**”, followed by the variable for the answer allows the user to input the answer to the True or False Questions. Sixthly, a decision structure decides on the correct answer for the answer variable. If the condition for the correct answer is true, the program window will clear the screen using “**cls**”. Then, the program window will display a message stating that the answer was correct using “put”. Next, the “**counter=counter+1**” will add 1 point to the counter, which counts how many correct answers that the user has. At the same time, the program will play an applause music file using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**” Also, an image corresponding to the question and an image displaying the word “Correct!” will be displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. However, if the answer is wrong, the program window will clear the screen using “**cls**”. Then, the program window will display a message stating that the answer was incorrect and will display the correct answer using “put”. Next, the program will play a music file with the sound of someone yelling, using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**”. Also, the program will display an image that displays the word “Incorrect!”, using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Lastly, the decision structure is ended using “**end if**”, delayed for 4 seconds using “**delay (4000)**”, and the screen is cleared using “**cls**”.

For each Fill in the Blanks Question, the following programming structure is used. Firstly, the variables for the answers and hints are declared as strings using “**var**”. Secondly, a music file is played using the “**Music.PlayFileLoop**” structure. Thirdly, the program window will display twelve blank spaces before the question using “**put””**”. Then the program window will display the question using “**put**”. Afterwards, the program window will display another four blank spaces using “**put””**”. Fourthly, an image that displays the title “Fill in the Blanks” is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Fifthly, a prompt message asking the user if they want a hint is displayed using “**put**”. Then, “**get**”, followed by the variable for the hint, allows the user to choose whether they want a hint or not. Sixthly, a decision structure is implemented – this decision structure will give the user a hint if their response was a “yes”. If the condition for the hint is true, then the program window will clear the screen using “**cls**”, display a hint message using “**put**”, and display an image that corresponds to the hint and an image the displays the word “HINT”, using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Then, there is a delay of 10 seconds using the “**delay (10000)**” structure. Following that, the program window will clear the screen. However, if the user chooses not to receive a hint, the program will immediately clear the screen using “**cls**”. Seventhly, the program window will display twelve blank spaces before the question using “**put””**”. Then the program window will display the question using “**put**”. Afterwards, the program window will display another four blank spaces using “**put””**”. Eighthly, an image that displays the title “Fill in the Blanks” is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Ninthly, the prompt message for inputting the answer is displayed using “**put**”. Afterwards, “**get**”, followed by the variable for the answer allows the user to input the answer to the Fill in the Blanks Question. Tenthly, a decision structure decides on the correct answer for the answer variable. If the condition for the correct answer is true, the program window will clear the screen using “**cls**”. Then, the program window will display a message stating that the answer was correct using “put”. Next, the “**counter=counter+1**” will add 1 point to the counter, which counts how many correct answers that the user has. At the same time, the program will play an applause music file using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**” Also, an image corresponding to the question and an image displaying the word “Correct!” will be displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. However, if the answer is wrong, the program window will clear the screen using “**cls**”. Then, the program window will display a message stating that the answer was incorrect and will display the correct answer using “put”. Next, the program will play a music file with the sound of someone yelling, using the “**Music.PlayFileLoop**” structure and stop the music using the “**Music.PlayFileStop**”. Also, the program will display an image that displays the word “Incorrect!” using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Lastly, the decision structure is ended using “**end if**”, delayed for 4 seconds using “**delay (4000)**”, and the screen is cleared using “**cls**”.

Programming Structure for Displaying Final Score:

The following will describe the programming structure for displaying the user’s final score. Firstly, a loop is implemented in the beginning to allow the user to return to the Final Score page. Secondly, a music file is played using the “**Music.PlayFileLoop**” structure. Thirdly, the variable “**end4**” is declared as string. This variable will be used to end the loop for the Final Score. Fourthly, “**drawfillbox(0,0,700,500,black)**”is used to display a black filled box from lower left corner coordinate (0,0) to top right corner coordinate (700,500). This black filled box is used to fill in remainder of the program window not occupied by the image that is displayed afterwards. Fifthly, a large image of a frog with the words “Final Score” are displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Sixthly, the user’s final score is displayed using “**put**”. Then, the user’s percentage mark is calculated by dividing the counter by 15 and multiplying it by 100. The calculation looks like this: “**mark:=counter/15\*100**”. Seventhly, a decision structure is implemented, which decides on the letter grade that the user should get depending on their percentage mark and displays a message corresponding to the percentage mark using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. If the user got 100%, the letter grade is an A+ and the user is congratulated on their achievement. If the user got 80% or over, the letter grade is an A, and the user is congratulated for scoring that mark. If the user got a mark of 70% or over and less than 80%, then the letter grade is a B, and the user is told that the mark is good. If the user got a mark of 60% or over and less than 70%, then the letter grade is a C, and the user is told to work harder. If the user’s mark is 50% or above and below 60%, the letter grade is a D, and the user is told to look through their science notes and retake the quiz again. If the user’s mark is below 50%, the letter grade is an R, and the user is told that they failed. After the program displays the marks the user got, the program is delayed for 6 seconds using “**delay (6000)**”. Then a prompt message, displayed using “**put**”, tells the user to input “yes” to continue the program. Inputting “yes” and pressing the enter key will terminate the loop for the Final Score page because “**exit when**” structure was implemented for the variable “**end4**”, where “**end4**” has to equal “yes”. Otherwise, inputting anything else and pressing enter would send the user back to the beginning of the Final Score Page. Lastly, at the end of the Final Score Page loop, there is the structure “**end if**”, which ends the decision structure for the mark, the structure “end loop”, which closes the loop for the Final Score page, and “**cls**”, which clears the screen and displays the prompt page for redoing the quiz.

Meanwhile, the following will describe the programming structure for obtaining the certificate. Firstly, the user must obtain a 100% mark in order to receive the certificate of accomplishment. Secondly, if the user receives a 100% mark on the quiz, the program will declare the variable “**perfect**” as a string, using “**var**”. Thirdly, a prompt message is displayed using “**put**”. Then, the program will use the “**get**” structure, followed by the variable “**perfect**”, to get the user to input their decision. Fourthly, a decision structure for the displaying of the certificate is implemented, where “**perfect**” has to equal “yes”. If the user inputs “yes”, then the program will clear the screen and display a message telling the user how to print – this message is displayed using “**put**”. The program displays this message for 7 seconds due to the “**delay (7000)**” structure. Then, the program clears the screen and displays the certificate using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. At the same time, the program locates a coordinate using the “**locatexy**” structure and inputs the user’s name using “**put**” at that specific coordinate. Then, after a delay of 10 seconds due to the structure “**delay (10000)**”, the user will be given the same prompt message telling the user to type yes to continue the program.

Programming Structure for the Prompt to Redo the Quiz:

The following describes the programming structure for the prompt, which asks the users if they want to redo the quiz or not. Firstly, the variable “**r**” is declared as a string using put “**var**”. Secondly, a prompt message is displayed using “put” for the user, telling the user to input “end” to end the quiz and anything else to redo the quiz. Thirdly, an image of a globe is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Meanwhile, a maple leaf is drawn using the structure “**drawfillmapleleaf(10,10,100,100,red)**”, followed by the structure for the maple leaf outline “**drawmapleleaf(10,10,100,100,black)**” Then, another image that displays the question “Do you want to redo the Science Quiz” is displayed using the “**Pic.FileNew**” and the “**Pic.Draw**” structures. Fourthly, the “**get**” structure, followed by the variable “**r**”, gets the user to input whether they want to redo the quiz, or to end it. If the response is “end”, then the structure “**exit when r=”end”**” will terminate the loop for the entire program and will end it. However, inputting anything else and pressing the enter key will send the user directly to the beginning of the quiz program.

Problems Encountered During the Programming Phase:

The following describes the problems that were encountered during the programming phase of the Science Quiz program. Firstly, one of the most troubling problems that were encountered was that the Turing program couldn’t find the music files or the image files that were to be included in the program. This frustrating matter took quite a while to solve. At the end, it was discovered that the program could only find these files if they were saved in the same directory with the Turing program. Secondly, another frustrating problem that was encountered during the programming phase was the factor that the image files appeared oversized when they were displayed on the program window. To solve this problem, the picture had to be made smaller manually. In this case, the Microsoft Office Picture Manager software was used to resize the images so that they would fit properly in the program window when the program was executed. Thirdly, one other problem encountered during the programming phase was the problems with looping. During the early stages of the programming phase, it was difficult figuring out how to make the program exit the loop. Exiting a loop was crucial for enabling the user to end the program if they wanted to at the end. At the end, this problem was solved through looking at the Turing Documentation and searching up exit statements for loops, which said the programming structure “**exit when**” had to be used to exit a loop. Fourthly, a minor problem encountered was that the program could not determine the user’s name at the end of the Science Quiz, which caused the program to display an error. At the end, the loop that was implemented before the prompt for entering the user’s name was removed to make it easier for the program to identify the user’s name. In the beginning, that loop was supposed to get the user to input their name again in the event that they spelled their name incorrectly. Lastly, another minor problem encountered during the programming phase was the compatibility issues with some of the image extensions and music file extensions. To solve these problems, images with an extension that was not JPEG were pasted onto Microsoft Office Powerpoint and the extension was changed by saving the image as a picture with a JPEG extension. Meanwhile, the music without the WAV extensions was imported to Sony Vegas Movie Studio 8. From there, the movie was rendered as a WAV sound file, which is fully compatible with the Turing environment. In conclusion, these were the problems that were encountered during the programming phase of the Science Quiz program.

Conclusion:

Primarily, this entire report was dedicated to describing the Science Quiz Program in its entirety. The program was created as a summative project, which would be handed into the teacher on December 20th, 2007. The project would be worth 10% of the final mark and was considered to be an exam project. The theme of the project was Grade 10 Science and its objective was to quiz users on their knowledge of Science at the Grade 10 level. The program itself implemented many loops, images, and decision structures to make it more creative and more complex. The certificate of accomplishment that was also included in the program and the hints included in the Fill in the Blanks questions were also used to add creativity and complexity to this Science Quiz program. Basically, this report describes the purpose for the creation of this summative project, as well as the process that was taken to put together this summative project, which resulted in the Science Quiz program