**Programming assignment, Report section.**

Before going in depth, this code, in simple terms is meant to create a loop requiring the correct password to be entered, to “escape” the loop and progress further in the program.

The code “string” is typically used to convert something into a string format and can also be classed as a sequence of characters. For example, in my code I converted a 0 into a string so it could be displayed into the textbox. Attempt is the variable for which a user will try to create a password. On the next line we see the “Mydialogs.inputbox” this code is used for opening a box which allows the user to type numeric or alphabetic characters into. Followed by that line, it says “enter your password” This is to display a box clearly displaying what the text box is for. It’s functional as it clearly shows the user what to do. Additionally, My Dialogs was created and contains various pieces of functional code to use.

The next section is where the loop is basically created. The “while” function creates essentially a “ring-road” loop with only one answer. It can also be classified “as long as”. Therefore, by entering the next part “attempt! = basic” this is telling the program simply, if “basic” is not entered the program will keep looping and follow the next section of code. Next you can see the brackets. For example, “ }” is reached, the program goes back up to the code and tests the condition again to see if it’s correct. (This time we may get a different result if the user enters the correct password!)

Next two brackets are displayed so a message box can be shown. This section can also be classified as the middle of the loop. This is why you see the “{“brackets to contain the loop and encapsulate the main body(middle of the code). If the correct password is entered, it would be classified as a TRUE condition, which basically means the middle part of the code, closed with “{“would only be executed once. A message box is simply a box that pops up when a certain action is done for example, clicking on an image as done in my code. It’s functional as the message box can alert the user when something is correct, or a password has been successfully entered, for example. The message box states “wrong! – try again.” Which would only display when the correct password has not been entered. This is then followed by a semi-colon, to separate the lines of codes. Next to that, another attempt command with a dialog box is entered. The reason it’s been entered twice and is in the middle of the loop is because, without it the “attempt” value may never change which wouldn’t be functional as the user would not be able to continue with the programme thus being stuck in an endless loop. Finally, after a semicolon and brackets a message box which states “welcome to the program” This is for when the correct password is typed (which is basic!) the message will appear clearly informing the user the correct password is entered, and they can continue, and the condition of the code is now set to TRUE. This is functional, as without a welcome message, the user may be confused as to whether he got into the program or didn’t.

Overall, the entire piece of code is a “ring-road” loop or a WHILE loop which is very functional and is a key piece of code which allows most password entering software to operate. It’s integral and without it, passwords would never be the same.