Unit 12
The Characteristics and Uses of a Software Program

BY SOPHIE MAY

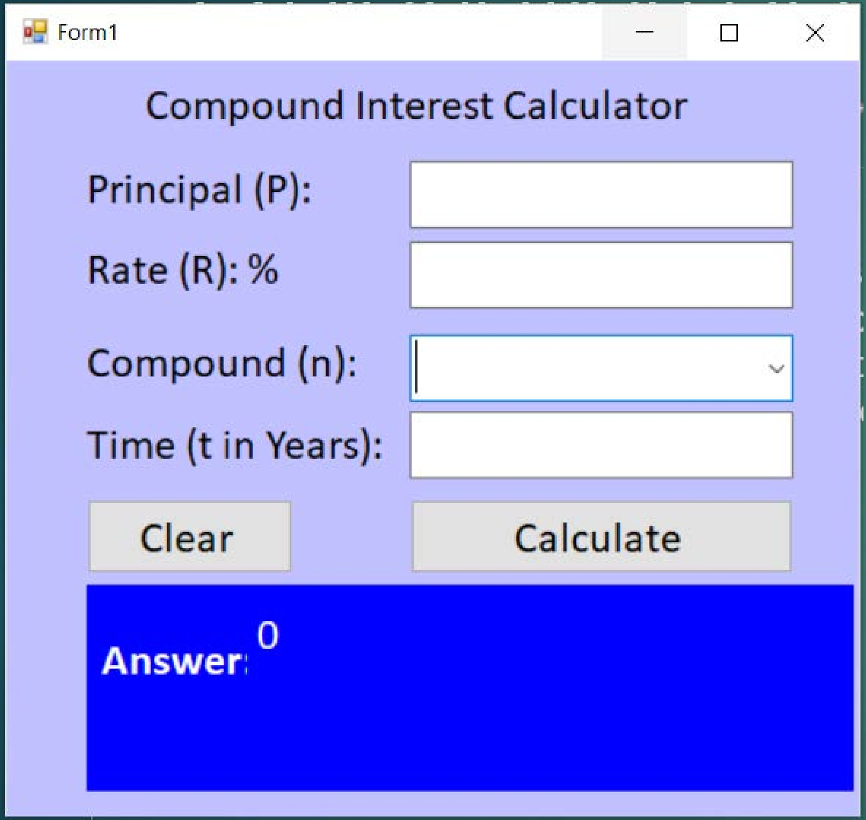
Introduction

This presentation has been created to provide an insight into the characteristics and uses of a software program. Within this presentation, I have provided examples of the key commands and structures, which can be manipulated and used to create a software program.

In order to demonstrate this, I have isolated the composition of code from two different software programs, to walk you through how each program has been developed and how each program can be improved.

The two programs I have demonstrated as evidence for this presentation are fairly basic functioning software programs, so they will help display the key functions and structures of a software program.

Explanation of Program One



Compound Interest

The first program used in this presentation is the compound interest calculator. The purpose of this program is to calculate the compound interest of a principal amount based on a specified growth timeframe, the compounding interval and rate of interest.

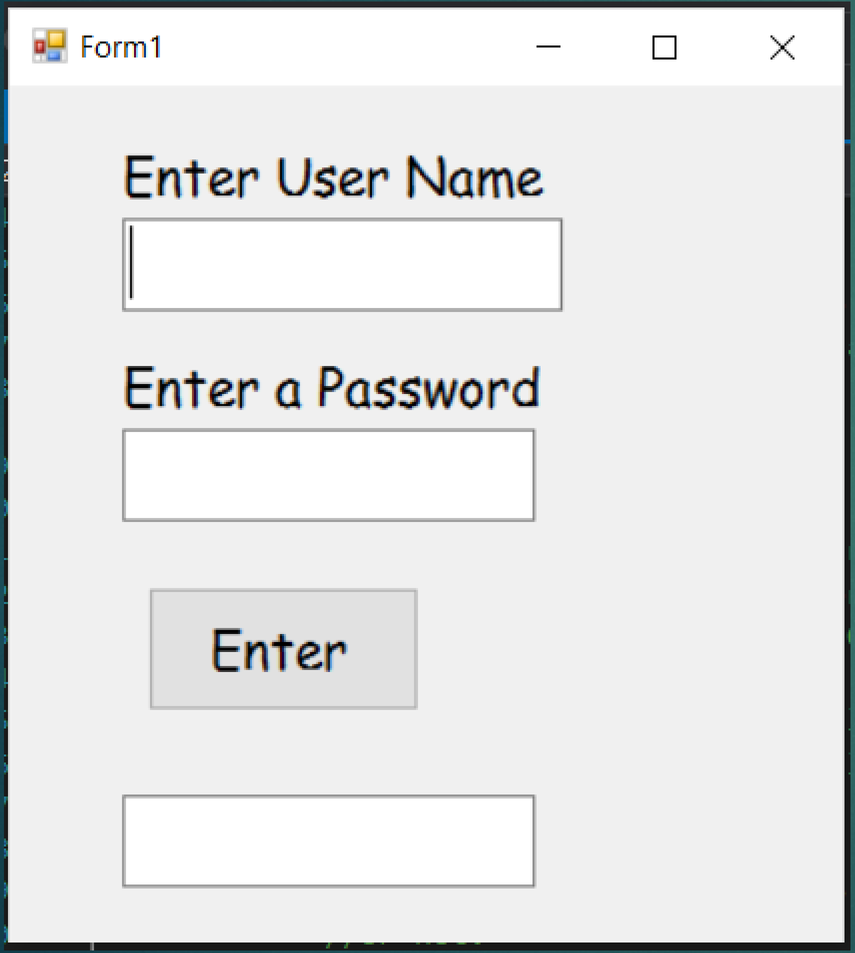
To use the calculator, the user is required to enter the principal amount in the principal amount text box (labelled “principal (p)”). Then the user must enter the interest rate in the interest in the rate text box.

Then the user selects the compounding interval from the drop down list on the combo box. Next the user enters a timeframe in years (10 equals 10 years), in the time text box.

Finally, the user must press the calculate button so the program can calculate the compound interest based on the data entered by the user.

Additionally, the program features a clear button to remove the data entered by the user, ready for the next user.

Explanation of Program Two



Unit 12 Passwords - Login Program

The second program used in this presentation is the unit 12 passwords login program. The purpose of this program is enable the user to login with a username and password.

To use the program, the user is required to enter their username into the “Enter User Name” text box. Then the user must enter their password into the “Enter a Password” text box.

Finally, the user must press the enter button so that the program can validate the data entered by the user against the data stored within the program.

Additionally, the program features a message box that allows the program to display a message to the user based on the validation of the data they entered. If the data is valid, the message reads “You can login”. However, if the data is invalid the message reads “You cannot login”.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in

txtbox.Text	username = txtUsername.Text;
	txtLogin.Text = "you cannot login";
		'.Text' is a command that reads from right to left and has two roles which are 'Get Text' (Right) and 'Set Text' (Left).
		The 'txtLogin.Text' is assigned to the left of the operator (=) which means that is where the program needs to send the data. This "sets" the data ("you cannot login"), and displays the data in the textbox labelled txtLogin.
		The 'txtUsername.Text' is assigned to the right of the assign operator (=) which means that is where the program needs to take the data from. This is where the program "gets" the data and stores it into the variable called "username".
			This command is found in both the login program called "Unit 12 Passwords" program and the "compound interest" program.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in
Convert.ToDecimal	princpal = Convert.ToDecimal(txtPrincipal.Text);	This command converts data from one datatype into a decimal datatype. So for this program, the purpose of the convert.ToDecimal command is used to instruct the program to convert the data stored in the variable called “principal”, into a decimal datatype.	This command is used in the compound interest program.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in
ToString	lblResult.Text = lblResult.Text + ("rate " + rate + " time " + time + " result " + result.ToString("#.##") + "\n\r");	The ToString command converts the result back from a decimal datatype into string data to be displayed in the output result text box.	This is found in the compound interest program.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in

=
Known as the assign operator
	decimal princpal = 0;
	decimal rate = 0;
	decimal compound = 1;
	int time = 0;
	decimal result = 0;
	lblResult.Text = "";
	int num =(int) cmbCompound.SelectedIndex;
	princpal = Convert.ToDecimal(txtPrincipal.Text);
	rate = Convert.ToDecimal(txtRate.Text)/100; ;
	time = Convert.ToInt16(txtTime.Text);
	txtLogin.Text = "you cannot login";
	txtLogin.Text = "You can login";
	username = txtUsername.Text;
	password = txtPassword.Text;
		The equal symbol is the assign operator which means that "something is assigned to something else".
		Such as in the examples I have included the first one says “decimal principal = 0”, which means that the variable principal has a decimal datatype and is assigned a value of 0.
		“princpal = Convert.ToDecimal(txtPrincipal.Text); ” means that the Convert.ToDecimal(txtPrincipal.Text) is assigned to the “principal” variable.
			The assign operator is found in both the compound interest and password program.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in

If statements
 	
(known as a decision statement, or a conditional statement)
	int num =(int) cmbCompound.SelectedIndex;
	if (num == 0) {
	} else if (num == 1)
	{
	compound = 12;
	}
	else if (num == 2)
	{
	compound = 2;
	}
	else if (num == 3)
	{
	compound = 1;
	
	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
	txtLogin.Text = "You can login";
	break;
	else
	{
	txtLogin.Text = "you cannot login";
	}
	count++;
	}
		The conditional if and else statements are used in a program to make a decision, it is a conditional instruction for the program to complete an action if certain conditions are met, else the program will be asked to do something else.
		For example, if the number in the compound interest combo-box is equal to 1, the compound interest is assigned a value of 12. Else if the number is equal to 2, then the compound interest is assigned a value of 2. Else if the number is equal to 3, then the compound interest is assigned a value of 1.
		
		On the password program the if and else statements are used to validate the data entered by the user. The if statement allows the program to verify that if the data entered into the username variable equals any of the data contained in column 0 and the data entered into the password variable equals any of the corresponding data contained in column 1, then the message displayed to the user is “You can login”.
		
		Else, the message displayed to the user is “you cannot login” and adds a value of 1 to the count variable.
			These conditional if and else statements are found in both the compound interest program and the password program.

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in

Variables	decimal princpal = 0;
	decimal rate = 0;
	decimal compound = 1;
	int time = 0;
	
	int count = 0;
	string username = "";
	string password = "";
		Variables are used to store data, we have to declare variables to enable the program to retrieve the stored data. A variable has a datatype to declare the size of the variable and the type of data that can be stored inside the variable.
		This enables the program to separate the data and validate the data and display it in a specific way to the user.
		
		For example, the variable “principal” in the compound interest program is assigned a value of 0 and a data type of decimal which means the size of the variable is 8 bytes.
			Variables are found in both the compound interest program and the password program.

Key Commands and Tools

<table>
<tr><th>Command</th><th>Example of Use</th><th>What does the code do within the program</th><th>Which Program is it found in</th></tr>
<tr>
<td><b>Methods</b></td>
<td>

```
public void storeData() {
princpal = Convert.ToDecimal(txtPrincipal.Text);
rate = Convert.ToDecimal(txtRate.Text)/100; ;
time = Convert.ToInt16(txtTime.Text);}

public void calcInterest() {
decimal result = 0;
lblResult.Text = "";
for (int t = 1; t < time + 1; t++) {
result =  princpal * (decimal) Math.Pow((double)( 1 +  rate / compound),
(double) (compound * t));
}
lblResult.Text = lblResult.Text + ("rate " + rate + " time " + time + "
result " + result.ToString("#.##") + "\n\r");}

public void chkLogins()
{
int count = 0;
while (count < nameArray.GetUpperBound(0))
{
if (username.Equals(nameArray[count,0]) &&
password.Equals(nameArray[count,1]))
{
txtLogin.Text = "You can login";
break;
}
else
{
txtLogin.Text = "you cannot login";
}
count++;}
```

</td>
<td>

A method is a group of instructions or tasks that are used to perform an action.

This makes it much easier for developers to be able to locate, identify and fix problems or make improvements to the program by combing through specific areas of the code.

Using methods to group instructions together also makes the code look much neater and concise which is handy for programs that contain hundreds of lines of code.

For example, the chkLogins() method contains the conditional statements pertaining to the log in conditions of the program.

If there was a problem with the log in validation process in the program, the developer could log the area where the program is struggling or crashing and pinpoint the exact line of code that is causing the program to incorrectly validate the log in details.

Methods allow the developers to section out areas of the code, which allows themselves and other developers to organise the code to an industry standard.

</td>
<td>Methods can be found in both the compound interest program and the password log in program.</td>
</tr>
</table>

Key Commands and Tools

Command	Example of Use	What does the code do within the program	Which Program is it found in

.Equals	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
		The .Equals command is used to compare two objects or values, to check that both objects and values are equal to each other.
		In the example of use that I have provided, the if statement is used to condition the program to validate the user's entered data against the username and password data stored within the program.
		If the username matches data from column 0 in the array and the password matches the corresponding data from column 1, then the user is permitted login access and a message is displayed to tell the user they can login. Thereby, checking that data is equal to the data stored in the program.
			This command is only used in the password login program.

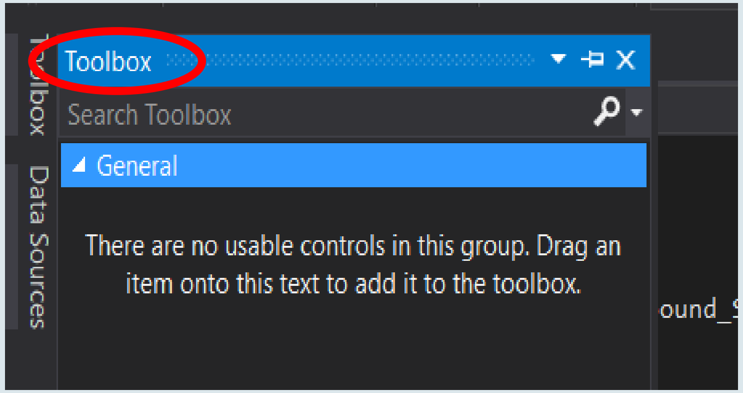
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in
IDE	Visual Studio	IDE stands for Integrated Development Environment. An IDE is an application suite used to make software programs. It uses a combination of integrated tools to aid developers in the production of software. The integrated tools available are the editing suite, the debugging tool and the compiling of the code created to machine code.	Both programs (Compound interest and password login) were created using visual studios but using different version. The compound interest calculator was made using the 2015 edition. However, the password login program was made using the 2010 shell edition.

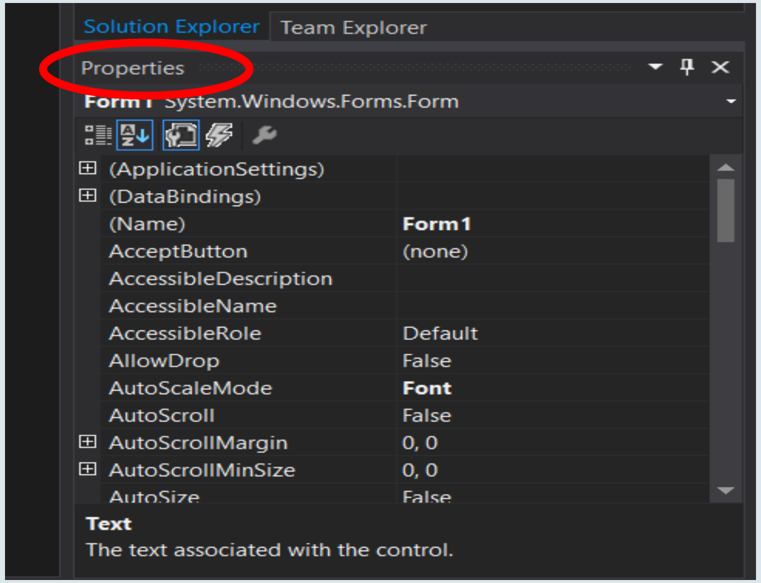
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in
IDE	Visual Studio	IDE stands for Integrated Development Environment. An IDE is an application suite used to make software programs. It uses a combination of integrated tools to aid developers in the production of software. The integrated tools available are the editing suite, the debugging tool and the compiling of the code created to machine code.	Both programs (Compound interest and password login) were created using visual studios but using different version. The compound interest calculator was made using the 2015 edition. However, the password login program was made using the 2010 shell edition.

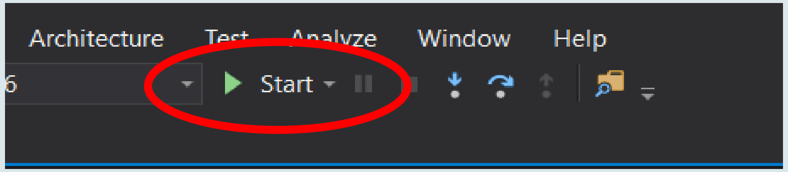
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in
Toolbox	 Toolbox Data Sources Toolbox Search Toolbox General ound_ There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.	The toolbox displays a selection of icons and controls which allows the developer to create actions within these controls. These icons and controls are suggested to the developer based on the type of file the developer is working on, this ensures the suitability of the controls to the program.	This tool is found in the visual studios IDE.

Key Commands and Tools

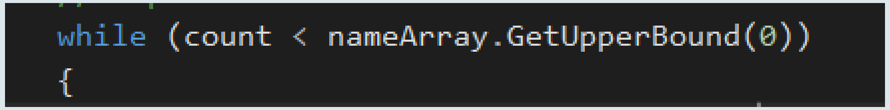
Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in
Properties box	 Form1 System.Windows.Forms.Form (ApplicationSettings) (DataBindings) (Name) **Form1** AcceptButton (none) AccessibleDescription AccessibleName AccessibleRole Default AllowDrop False AutoScaleMode **Font** AutoScroll False AutoScrollMargin 0, 0 AutoScrollMinSize 0, 0 AutoSize False **Text** The text associated with the control.	This tool is available in the visual studio IDE, and is used to alter the design properties and events within a program.	This is tool available for use on visual studios and both programs can access this function.

Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in
Run Button		The run button is available to run the program to allow the developer to view the program they have created. This allows them to check for bugs and fix them and check that the program has been correctly coded.	This is a tool available in the visual studios software, and both programs can use this tool.

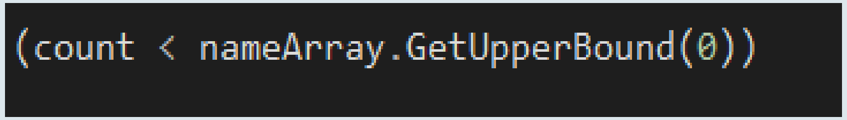
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

While Loop	while (count < nameArray.GetUpperBound(0))
	{
	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
	txtLogin.Text = "You can login";
	break;}
	else
	{
	txtLogin.Text = "you cannot login";}
	count++;}}
	
	while (count < nameArray.GetUpperBound(0))
	{
		A while loop is a used to pinpoint a section or line of code and repeat it on a continual loop based on, whether the conditional statements are returned true or false.
		In the password program, the while loop is shown in the code provided (left). The code shows the conditional statements while the loop is in place.
		The condition is checking to determine the length of the number of rows in the array.
		The program then dictates what it must do based on the conditional statements.
		For example, when the if statement is returned true, the while loop is within the bounds set by the GetUpperBounds keyword and the user is allowed to log in.
		Else if the statement is returned false, then the data entered by the user is outside the bounds and the else condition is executed.
			This keyword is only used in the password login program.

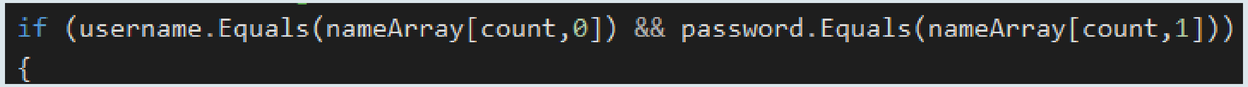
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

GetUpperBounds	while (count < nameArray.GetUpperBound(0))
	{
	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
	txtLogin.Text = "You can login";
	break;}
	else
	{
	txtLogin.Text = "you cannot login";}
	count++;}}
	
	(count < nameArray.GetUpperBound(0))
		The GetUpperBounds tells the program to count the number of rows in the array, which tells the program the upper limits of the array.
		This is used to make sure the conditions are followed based on the length of the array, and anything outside the length of the array follows the else conditional statement.
			This keyword command is found only in the password login program.

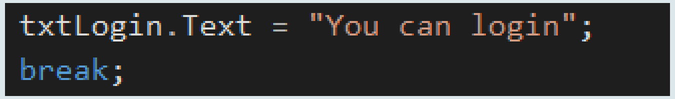
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

Logical Operator (&&)	while (count < nameArray.GetUpperBound(0))
	{
	
	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
	txtLogin.Text = "You can login";
	break;}
	
	else
	{
	txtLogin.Text = "you cannot login";}
	count++;}}
	
	if (username.Equals(nameArray[count,0]) && password.Equals(nameArray[count,1]))
	{
		This is a logical "And" operator. The && is used to check if both username and password returned true and if both are true then the message "you can login", is displayed to the user.	This is operator is used in the password login program.

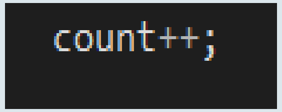
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

Break	while (count < nameArray.GetUpperBound(0))
	{
	
	if (username.Equals(nameArray[count,0]) &&
	password.Equals(nameArray[count,1]))
	{
	txtLogin.Text = "You can login";
	break;}
	
	else
	{
	txtLogin.Text = "you cannot login";}
	count++;}}
	
	txtLogin.Text = "You can login";
	break;
		The break statement is used to break the loop, to allow the program to execute the next statement after the loop.
		
		In this code, the break statement is executed after the if statement is returned true and the message is displayed to the user.
		
		The break statement is necessary to terminate the loop and allow the program to execute the next statement.
			The break statement is only found in the password login program.

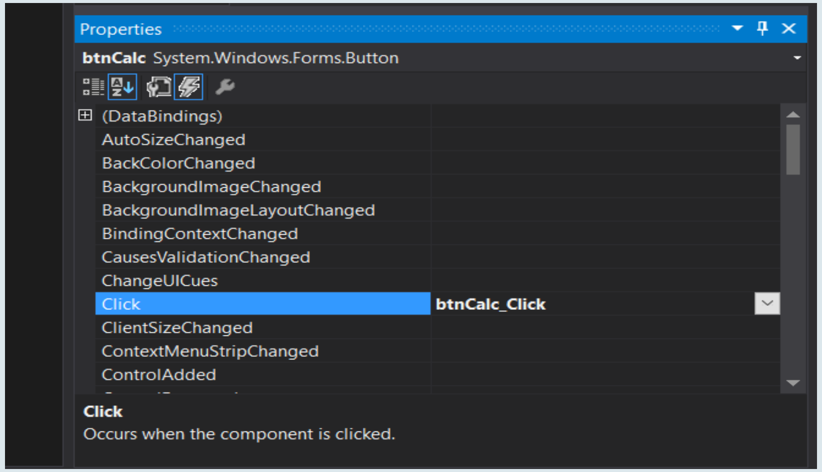
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

++	count++;}}
	
	public void calcInterest() {
	decimal result = 0; lblResult.Text = "";
	
	for (int t = 1; t < time + 1; t++) {
	result = princpal * (decimal)
	
	
	count++;
		This is an arithmetic operator that is used to increase the integer value by one.
		
		In the password login program the line of code that says count++; means that the program will add 1 to the value of the count variable when the conditional statement is returned false.
			This operator is found in both programs (compound interest and password login program).

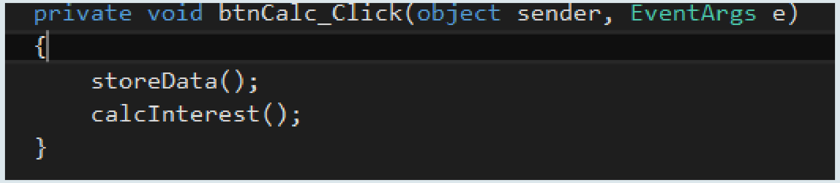
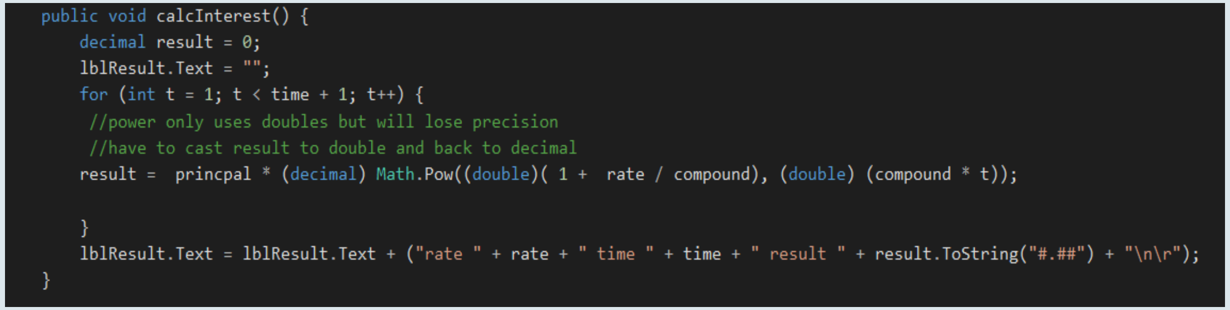
Key Commands and Tools

Command/Tool	Example of Use	What does the code do within the program	Which Program is it found in

Trigger	private void btnCalc_Click(object sender, EventArgs e)
	{
	storeData();
	calcInterest();}
	
	private void btnEnter_Click(object sender, EventArgs e)
	{
	username = txtUsername.Text; password = txtPassword.Text;
	chkLogins();}
	
	Properties
	btnCalc System.Windows.Forms.Button
	(DataBindings)
	AutoSizeChanged
	BackColorChanged
	BackgroundImageChanged
	BackgroundImageLayoutChanged
	BindingContextChanged
	CausesValidationChanged
	ChangeUICues
	Click **btnCalc_Click**
	ClientSizeChanged
	ContextMenuStripChanged
	ControlAdded
	Click
	Occurs when the component is clicked.
		A trigger is an event-driven procedure that initiates or "triggers" an action (event) to occur.
		
		This means that a procedure must be followed so that an action can be triggered to cause an event to run.
		
		For example, in order for the set of instructions contained in the event to run, an action such as a button being pressed must occur in order to "trigger", the event.
			A trigger can be found in both the compound interest program and the password login program.

Key Commands and Tools

Command/ Tool	Example of Use	What does the code do within the program	Which Program is it found in

Event	private void btnCalc_Click(object sender, EventArgs e)
	{
	storeData();
	calcInterest();}
	
	private void btnEnter_Click(object sender, EventArgs e)
	{
	username = txtUsername.Text; password = txtPassword.Text; chkLogins();}
	
	
	private void btnCalc_Click(object sender, EventArgs e)
	{
	    storeData();
	    calcInterest();
	}
	
	
	public void calcInterest() {
	    decimal result = 0;
	    lblResult.Text = "";
	    for (int t = 1; t < time + 1; t++) {
	     //power only uses doubles but will lose precision
	     //have to cast result to double and back to decimal
	    result = princpal * (decimal) Math.Pow((double)(1 + rate / compound), (double) (compound * t));
	
	    }
	    lblResult.Text = lblResult.Text + ("rate " + rate + " time " + time + " result " + result.ToString("#.##") + "\n\r");
	}
		An event happens when an action occurs.
		
		For example, to get the program to process the compound interest calculation, the user must press a button to activate the calculation event.
		
		The event must be in place, so that the program can carry out the action of calculating the compound interest for the user.
		
		If the event is not in place, then the program will not calculate the compound interest and the program will be useless and unfit for purpose.
			An event can be found in both the compound interest program and the password login program.