

Namespace ASE_Assignment

Classes

[AppCanvas](#)

class for my canvas which will be drawn on, extending the Icanvas interface

[AppCircle](#)

Represents a command to draw a circle on the canvas.

[AppCommandFactory](#)

My CommandFactory which extends the original BOOSE.CommandFactory from the BOOSE library, which allows me to replace and add my own commands.

[AppDrawTo](#)

Represents a command for drawing a line on the canvas from the current pen position to the coordinates inputted by the user. This class inherits BOOSE.CommandTwoParameters and implements the BOOSE.ICommand interface.

[AppMoveTo](#)

Represents a command to move the position of the pen on the canvas. This class inherits BOOSE.CommandTwoParameters and implements BOOSE.ICommand.

[AppPenColour](#)

Represents a command to set the colour of the pen using RGB values. This class inherits BOOSE.CommandThreeParameters and implements BOOSE.ICommand.

[AppRect](#)

Represents a command for drawing a rectangle on the canvas.

[Form1](#)

Main form of program, provides UI for inputting and visualising user commands.

Class AppCanvas

Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

class for my canvas which will be drawn on, extending the Icanvas interface

```
public class AppCanvas : ICanvas
```








Inheritance

[object](#)  ← [AppCanvas](#)

Implements

ICanvas

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Constructors

AppCanvas()

initializes a new instance of AppCanvas class and sets default size of canvas

```
public AppCanvas()
```

Properties

PenColour

gets the colour of the pen

```
public object PenColour { get; set; }
```

Property Value

[object](#)

Xpos

sets the X coordinate of the pen position on the canvas

```
public int Xpos { get; set; }
```

Property Value

[int](#)

Ypos

gets the y coordinate of the pen position on the canvas

```
public int Ypos { get; set; }
```

Property Value

[int](#)

Methods

Circle(int, bool)

draws a circle onto the canvas at the current pen position

```
public void Circle(int radius, bool filled = false)
```

Parameters

radius [int](#)

radius of circle

filled [bool](#)

states whether circle is filled (true) or not (false)

Exceptions

CanvasException

exception thrown when radius of circle is negative

Clear()

clears the canvas by filling it with background colour

```
public void Clear()
```

DrawTo(int, int)

draws a line from current pen position to inputted coordinate on canvas

```
public void DrawTo(int xDraw, int yDraw)
```

Parameters

xDraw [int](#)

x coordinate of end point

yDraw [int](#)

y coordinate of end point

Exceptions

CanvasException

exception thrown when coordinates inputted are outside of canvas

GetPenColourComponents()

Retrieves the individual RGB components of the current pen colour.

```
public (int Red, int Green, int Blue) GetPenColourComponents()
```

Returns

([int](#) [Red](#), [int](#) [Green](#), [int](#) [Blue](#))

A tuple containing the red, green, and blue components of the pen color as integers from 0 to 255:

- **Red** -
- **Green** -
- **Blue** -

Remarks

This method is an easy way to retrieve the individual RGB components of the pen colour. Mainly used in Unit Testing.

MoveTo(int, int)

moves pen to inputted coordinates without drawing a line

```
public void MoveTo(int x, int y)
```

Parameters

x [int](#)

x coordinate of new position

y [int](#)

y coordinate of new position

Exceptions

CanvasException

exception thrown when coordinates outside of canvas are inputted

Rect(int, int, bool)

draws a rectangle on canvas at pen position

```
public void Rect(int width, int height, bool filled)
```

Parameters

width [int](#)

width of rectangle

height [int](#)

height of rectangle

filled [bool](#)

states whether rectangle is filled (true) or not (false)

Exceptions

CanvasException

exception thrown when the width and height are negative or equal

Reset()

resets pen position to (0,0)

```
public void Reset()
```

Set(int, int)

sets dimensions of canvas

```
public void Set(int XSIZE, int YSIZE)
```

Parameters

XSIZE [int](#)

width of canvas

YSIZE [int](#)

height of canvas

SetColour(int, int, int)

sets colour of pen using RGB values

```
public void SetColour(int red, int green, int blue)
```

Parameters

red [int](#)

the amount of red used in the colour of the pen out of 255

green [int](#)

the amount of green used in the colour of the pen out of 255

blue [int](#)

the amount of blue used in the colour of the pen out of 255

Exceptions

CanvasException

exception thrown when a colour value exceeds 255

Tri(int, int)

draws triangle on canvas

```
public void Tri(int width, int height)
```

Parameters

width [int](#)

width of triangle

height [int](#)

height of triangle

Exceptions

[NotImplementedException](#)

not yet implemented

WriteText(string)

writes text onto canvas

```
public void WriteText(string text)
```

Parameters

text [string](#)

the text which is displayed onto the canvas

Exceptions

[NotImplementedException](#)

not implemented yet

getBitmap()

returns bitmap used to render drawing

```
public object getBitmap()
```

Returns

[object](#)

bitmap of canvas

Class AppCircle


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Represents a command to draw a circle on the canvas.

```
public class AppCircle : CommandTwoParameters, ICommand
```








Inheritance

[object](#)  ← [Command](#) ← [CanvasCommand](#) ← [CommandOneParameter](#) ← [CommandTwoParameters](#) ← [AppCircle](#)

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
CommandTwoParameters.CheckParameters(string[]) , CommandOneParameter.param1 ,
CommandOneParameter.param1unprocessed , CommandOneParameter.CheckParameters(string[]) ,
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,
Command.CheckParameters(string[]) , Command.Set(StoredProgram, string) , Command.Compile() ,
Command.Execute() , Command.ProcessParameters(string) , Command.ToString() , Command.Program ,
Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint ,
[object.Equals\(object?\)](#)  , [object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Methods

Execute()

Executes the circle drawing command.

```
public override void Execute()
```

Remarks

This method retrieves the radius of the circle from user input. It then determines if circle should be filled or not based on provided parameters. It then invokes the method to draw the circle onto the canvas.

Class AppCommandFactory


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

My CommandFactory which extends the original BOOSE.CommandFactory from the BOOSE library, which allows me to replace and add my own commands.

```
public class AppCommandFactory : CommandFactory, ICommandFactory
```








Inheritance

[object](#)  ← [CommandFactory](#) ← [AppCommandFactory](#)

Implements

ICommandFactory

Inherited Members

[CommandFactory.MakeCommand\(string\)](#) , [object.Equals\(object?\)](#)  , [object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Constructors

AppCommandFactory()

Initializes a new instance of the ASE_Assignment.AppCommandFactory class.

```
public AppCommandFactory()
```

Methods

MakeCommand(string)

Creates a command based on the command type specified.

```
public override ICommand MakeCommand(string commandType)
```

Parameters

commandType [string](#) 

The type of the command to be created. This string is trimmed and converted to lowercase to ensure it works with the BOOSE parser..

Returns

ICommand

An instance of the corresponding BOOSE.ICommand implementation.

Remarks

If the command is included in my ASE_Assignment.AppCommandFactory, for example the "drawto" command, the parser will return my ASE_Assignment.AppDrawTo command instead of the original one. For other command types, the method delegates to the original BOOSE.CommandFactory to create the command.

Class AppDrawTo


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Represents a command for drawing a line on the canvas from the current pen position to the coordinates inputted by the user. This class inherits BOOSE.CommandTwoParameters and implements the BOOSE.ICommand interface.

```
public class AppDrawTo : CommandTwoParameters, ICommand
```








Inheritance

[object](#)  ← [Command](#) ← [CanvasCommand](#) ← [CommandOneParameter](#) ← [CommandTwoParameters](#) ← [AppDrawTo](#)

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
CommandTwoParameters.CheckParameters(string[]) , CommandOneParameter.param1 ,
CommandOneParameter.param1unprocessed , CommandOneParameter.CheckParameters(string[]) ,
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,
Command.CheckParameters(string[]) , Command.Set(StoredProgram, string) , Command.Compile() ,
Command.Execute() , Command.ProcessParameters(string) , Command.ToString() , Command.Program ,
Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint ,
[object.Equals\(object?\)](#)  , [object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Methods

Execute()

Executes my AppDrawTo command by parsing the parameters to integers, and making sure the canvas is retrieved.

```
public void Execute()
```

Class AppMoveTo


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Represents a command to move the position of the pen on the canvas. This class inherits `BOOSE.CommandTwoParameters` and implements `BOOSE.ICommand`.

```
public class AppMoveTo : CommandTwoParameters, ICommand
```








Inheritance

[object](#)  ← [Command](#) ← [CanvasCommand](#) ← [CommandOneParameter](#) ← [CommandTwoParameters](#) ← [AppMoveTo](#)

Implements

`ICommand`

Inherited Members

`CommandTwoParameters.param2` , `CommandTwoParameters.param2unprocessed` , `CommandTwoParameters.CheckParameters(string[])` , `CommandOneParameter.param1` , `CommandOneParameter.param1unprocessed` , `CommandOneParameter.CheckParameters(string[])` , `CanvasCommand.yPos` , `CanvasCommand.xPos` , `CanvasCommand.canvas` , `CanvasCommand.Canvas` , `Command.program` , `Command.parameterList` , `Command.parameters` , `Command.paramsint` , `Command.CheckParameters(string[])` , `Command.Set(StoredProgram, string)` , `Command.Compile()` , `Command.Execute()` , `Command.ProcessParameters(string)` , `Command.ToString()` , `Command.Program` , `Command.Name` , `Command.ParameterList` , `Command.Parameters` , `Command.Paramsint` , [object.Equals\(object?\)](#)  , [object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Methods

Execute()

Executes the command to move the drawing cursor to the specified (x, y) coordinates.


```
public override void Execute()
```

Remarks

The command parses the x and y values inputted by the user then moves the pen to the position on the canvas. It ensures that the provided coordinates are within the bounds of the canvas.

Class AppPenColour


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Represents a command to set the colour of the pen using RGB values. This class inherits BOOSE.CommandThreeParameters and implements BOOSE.ICommand.

```
public class AppPenColour : CommandThreeParameters, ICommand
```








Inheritance

[object](#)  ← [Command](#) ← [CanvasCommand](#) ← [CommandOneParameter](#) ← [CommandTwoParameters](#) ← [CommandThreeParameters](#) ← [AppPenColour](#)

Implements

ICommand

Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed ,
CommandThreeParameters.CheckParameters(string[]) , CommandTwoParameters.param2 ,
CommandTwoParameters.param2unprocessed , CommandTwoParameters.CheckParameters(string[]) ,
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,
CommandOneParameter.CheckParameters(string[]) , CanvasCommand.yPos , CanvasCommand.xPos ,
CanvasCommand.canvas , CanvasCommand.Canvas , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , Command.CheckParameters(string[]) ,
Command.Set(StoredProgram, string) , Command.Compile() , Command.Execute() ,
Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object?\)](#)  ,
[object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Methods

Execute()

Executes the command to set the pen color based on RGB values.

```
public override void Execute()
```

Remarks

The command expects three parameters: red, green, and blue, each representing a color component. The method ensures that the RGB values are valid (between 0 and 255) and calculates the correct colour based on the values.

Class AppRect


Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Represents a command for drawing a rectangle on the canvas.

```
public class AppRect : CommandThreeParameters, ICommand
```








Inheritance

[object](#)  ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← CommandThreeParameters ← [AppRect](#)

Implements

ICommand

Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed ,
CommandThreeParameters.CheckParameters(string[]) , CommandTwoParameters.param2 ,
CommandTwoParameters.param2unprocessed , CommandTwoParameters.CheckParameters(string[]) ,
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,
CommandOneParameter.CheckParameters(string[]) , CanvasCommand.yPos , CanvasCommand.xPos ,
CanvasCommand.canvas , CanvasCommand.Canvas , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , Command.CheckParameters(string[]) ,
Command.Set(StoredProgram, string) , Command.Compile() , Command.Execute() ,
Command.ProcessParameters(string) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object?\)](#)  ,
[object.Equals\(object?, object?\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object?, object?\)](#)  , [object.ToString\(\)](#) 

Remarks

This class inherits BOOSE.CommandThreeParameters and implements the BOOSE.ICommand interface. It enables the drawing of a rectangle by specifying its width, height, and whether it should be filled.

Methods

Execute()

Executes the rectangle drawing command.

```
public override void Execute()
```

Remarks

This method retrieves the rectangles dimensions from user input and determines whether the rectangle should be filled or unfilled, based on the provided parameters. It then invokes the method to draw the rectangle on the canvas.

Class Form1

Namespace: [ASE Assignment](#)

Assembly: BOOSEproject.dll

Main form of program, provides UI for inputting and visualising user commands.

```
public class Form1 : Form, IDropTarget, ISynchronizeInvoke, IWin32Window,
    IBindableComponent, IComponent, IDisposable, IContainerControl
```

Inheritance

[object](#) ← [MarshalByRefObject](#) ← [Component](#) ← [Control](#) ← [ScrollableControl](#) ← [ContainerControl](#) ← [Form](#) ← [Form1](#)

Implements

[IDropTarget](#) , [ISynchronizeInvoke](#) , [IWin32Window](#) , [IBindableComponent](#) , [IComponent](#) , [IDisposable](#) , [IContainerControl](#)

Inherited Members

[Form.SetVisibleCore\(bool\)](#) , [Form.Activate\(\)](#) , [Form.ActivateMdiChild\(Form?\)](#) , [Form.AddOwnedForm\(Form?\)](#) , [Form.AdjustFormScrollbars\(bool\)](#) , [Form.Close\(\)](#) , [Form.CreateAccessibilityInstance\(\)](#) , [Form.CreateControlsInstance\(\)](#) , [Form.CreateHandle\(\)](#) , [Form.DefWndProc\(ref Message\)](#) , [Form.Dispose\(bool\)](#) , [Form.ProcessMnemonic\(char\)](#) , [Form.CenterToParent\(\)](#) , [Form.CenterToScreen\(\)](#) , [Form.LayoutMdi\(MdiLayout\)](#) , [Form.OnActivated\(EventArgs\)](#) , [Form.OnBackgroundImageChanged\(EventArgs\)](#) , [Form.OnBackgroundImageLayoutChanged\(EventArgs\)](#) , [Form.OnClosing\(CancelEventArgs\)](#) , [Form.OnClosed\(EventArgs\)](#) , [Form.OnFormClosing\(FormClosingEventArgs\)](#) , [Form.OnFormClosed\(FormClosedEventArgs\)](#) , [Form.OnCreateControl\(\)](#) , [Form.OnDeactivate\(EventArgs\)](#) , [Form.OnEnabledChanged\(EventArgs\)](#) , [Form.OnEnter\(EventArgs\)](#) , [Form.OnFontChanged\(EventArgs\)](#) , [Form.OnGotFocus\(EventArgs\)](#) , [Form.OnHandleCreated\(EventArgs\)](#) , [Form.OnHandleDestroyed\(EventArgs\)](#) , [Form.OnHelpButtonClicked\(CancelEventArgs\)](#) , [Form.OnLayout\(LayoutEventArgs\)](#) , [Form.OnLoad\(EventArgs\)](#) , [Form.OnMaximizedBoundsChanged\(EventArgs\)](#) , [Form.OnMaximumSizeChanged\(EventArgs\)](#) , [Form.OnMinimumSizeChanged\(EventArgs\)](#) , [Form.OnInputLanguageChanged\(InputLanguageChangedEventArgs\)](#) , [Form.OnInputLanguageChanging\(InputLanguageChangingEventArgs\)](#) , [Form.OnVisibleChanged\(EventArgs\)](#) , [Form.OnMdiChildActivate\(EventArgs\)](#) ,

[Form.OnMenuStart\(EventArgs\)](#), [Form.OnMenuComplete\(EventArgs\)](#),
[Form.OnPaint\(PaintEventArgs\)](#), [Form.OnResize\(EventArgs\)](#),
[Form.OnDpiChanged\(DpiChangedEventArgs\)](#), [Form.OnGetDpiScaledSize\(int, int, ref Size\)](#),
[Form.OnRightToLeftLayoutChanged\(EventArgs\)](#), [Form.OnShown\(EventArgs\)](#),
[Form.OnTextChanged\(EventArgs\)](#), [Form.ProcessCmdKey\(ref Message, Keys\)](#),
[Form.ProcessDialogKey\(Keys\)](#), [Form.ProcessDialogChar\(char\)](#),
[Form.ProcessKeyPreview\(ref Message\)](#), [Form.ProcessTabKey\(bool\)](#),
[Form.RemoveOwnedForm\(Form?\)](#), [Form.Select\(bool, bool\)](#),
[Form.ScaleMinMaxSize\(float, float, bool\)](#),
[Form.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#),
[Form.ScaleControl\(SizeF, BoundsSpecified\)](#), [Form.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#),
[Form.SetClientSizeCore\(int, int\)](#), [Form.SetDesktopBounds\(int, int, int, int\)](#),
[Form.SetDesktopLocation\(int, int\)](#), [Form.Show\(IWin32Window?\)](#), [Form.ShowDialog\(\)](#),
[Form.ShowDialog\(IWin32Window?\)](#), [Form.ToString\(\)](#), [Form.UpdateDefaultButton\(\)](#),
[Form.OnResizeBegin\(EventArgs\)](#), [Form.OnResizeEnd\(EventArgs\)](#),
[Form.OnStyleChanged\(EventArgs\)](#), [Form.ValidateChildren\(\)](#),
[Form.ValidateChildren\(ValidationConstraints\)](#), [Form.WndProc\(ref Message\)](#), [Form.AcceptButton](#),
[Form.ActiveForm](#), [Form.ActiveMdiChild](#), [Form.AllowTransparency](#), [Form.AutoScroll](#),
[Form.AutoSize](#), [Form.AutoSizeMode](#), [Form.AutoValidate](#), [Form.BackColor](#),
[Form.FormBorderStyle](#), [Form.CancelButton](#), [Form.ClientSize](#), [Form.ControlBox](#),
[Form.CreateParams](#), [Form.DefaultImeMode](#), [Form.DefaultSize](#), [Form.DesktopBounds](#),
[Form.DesktopLocation](#), [Form.DialogResult](#), [Form.HelpButton](#), [Form.Icon](#), [Form.IsMdiChild](#),
[Form.IsMdiContainer](#), [Form.IsRestrictedWindow](#), [Form.KeyPreview](#), [Form.Location](#),
[Form.MaximizedBounds](#), [Form.MaximumSize](#), [Form.MainMenuStrip](#), [Form.MinimumSize](#),
[Form.MaximizeBox](#), [Form.MdiChildren](#), [Form.MdiChildrenMinimizedAnchorBottom](#),
[Form.MdiParent](#), [Form.MinimizeBox](#), [Form.Modal](#), [Form.Opacity](#), [Form.OwnedForms](#),
[Form.Owner](#), [Form.RestoreBounds](#), [Form.RightToLeftLayout](#), [Form.ShowInTaskbar](#),
[Form.ShowIcon](#), [Form.ShowWithoutActivation](#), [Form.Size](#), [Form.SizeGripStyle](#),
[Form.StartPosition](#), [Form.Text](#), [Form.TopLevel](#), [Form.TopMost](#), [Form.TransparencyKey](#),
[Form.WindowState](#), [Form.AutoSizeChanged](#), [Form.AutoValidateChanged](#),
[Form.HelpButtonClicked](#), [Form.MaximizedBoundsChanged](#), [Form.MaximumSizeChanged](#),
[Form.MinimumSizeChanged](#), [Form.Activated](#), [Form.Deactivate](#), [Form.FormClosing](#),
[Form.FormClosed](#), [Form.Load](#), [Form.MdiChildActivate](#), [Form.MenuComplete](#),
[Form.MenuStart](#), [Form.InputLanguageChanged](#), [Form.InputLanguageChanging](#),
[Form.RightToLeftLayoutChanged](#), [Form.Shown](#), [Form.DpiChanged](#), [Form.ResizeBegin](#),
[Form.ResizeEnd](#), [ContainerControl.AdjustFormScrollbars\(bool\)](#), [ContainerControl.Dispose\(bool\)](#),
[ContainerControl.OnAutoValidateChanged\(EventArgs\)](#), [ContainerControl.OnCreateControl\(\)](#),
[ContainerControl.OnFontChanged\(EventArgs\)](#), [ContainerControl.OnLayout\(LayoutEventArgs\)](#),
[ContainerControl.OnMove\(EventArgs\)](#), [ContainerControl.OnParentChanged\(EventArgs\)](#),
[ContainerControl.OnResize\(EventArgs\)](#), [ContainerControl.PerformAutoScale\(\)](#),

[ContainerControl.ScaleMinMaxSize\(float, float, bool\)](#), [ContainerControl.ProcessDialogChar\(char\)](#), [ContainerControl.ProcessDialogKey\(Keys\)](#), [ContainerControl.ProcessCmdKey\(ref Message, Keys\)](#), [ContainerControl.ProcessMnemonic\(char\)](#), [ContainerControl.ProcessTabKey\(bool\)](#), [ContainerControl.RescaleConstantsForDpi\(int, int\)](#), [ContainerControl.Select\(bool, bool\)](#), [ContainerControl.UpdateDefaultButton\(\)](#), [ContainerControl.Validate\(\)](#), [ContainerControl.Validate\(bool\)](#), [ContainerControl.WndProc\(ref Message\)](#), [ContainerControl.AutoScaleDimensions](#), [ContainerControl.AutoScaleFactor](#), [ContainerControl.AutoScaleMode](#), [ContainerControl.BindingContext](#), [ContainerControl.CanEnableIme](#), [ContainerControl.ActiveControl](#), [ContainerControl.CreateParams](#), [ContainerControl.CurrentAutoScaleDimensions](#), [ContainerControl.ParentForm](#), [ScrollableControl.ScrollStateAutoScrolling](#), [ScrollableControl.ScrollStateHScrollVisible](#), [ScrollableControl.ScrollStateVScrollVisible](#), [ScrollableControl.ScrollStateUserHasScrolled](#), [ScrollableControl.ScrollStateFullDrag](#), [ScrollableControl.AdjustFormScrollbars\(bool\)](#), [ScrollableControl.GetScrollState\(int\)](#), [ScrollableControl.OnLayout\(LayoutEventArgs\)](#), [ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#), [ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#), [ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#), [ScrollableControl.OnPaddingChanged\(EventArgs\)](#), [ScrollableControl.OnVisibleChanged\(EventArgs\)](#), [ScrollableControl.ScaleControl\(SizeF, BoundsSpecified\)](#), [ScrollableControl.SetDisplayRectLocation\(int, int\)](#), [ScrollableControl.ScrollControlIntoView\(Control?\)](#), [ScrollableControl.ScrollToControl\(Control\)](#), [ScrollableControl.OnScroll\(ScrollEventArgs\)](#), [ScrollableControl.SetAutoScrollMargin\(int, int\)](#), [ScrollableControl.SetScrollState\(int, bool\)](#), [ScrollableControl.WndProc\(ref Message\)](#), [ScrollableControl.AutoScroll](#), [ScrollableControl.AutoScrollMargin](#), [ScrollableControl.AutoScrollPosition](#), [ScrollableControl.AutoScrollMinSize](#), [ScrollableControl.CreateParams](#), [ScrollableControl.DisplayRectangle](#), [ScrollableControl.HScroll](#), [ScrollableControl.HorizontalScroll](#), [ScrollableControl.VScroll](#), [ScrollableControl.VerticalScroll](#), [ScrollableControl.Scroll](#), [Control.GetAccessibilityObjectById\(int\)](#), [Control.SetAutoSizeMode\(AutoSizeMode\)](#), [Control.GetAutoSizeMode\(\)](#), [Control.GetPreferredSize\(Size\)](#), [Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#), [Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#), [Control.BeginInvoke\(Delegate\)](#), [Control.BeginInvoke\(Action\)](#), [Control.BeginInvoke\(Delegate, params object?\[\]\)](#), [Control.BringToFront\(\)](#), [Control.Contains\(Control?\)](#), [Control.CreateAccessibilityInstance\(\)](#), [Control.CreateControlsInstance\(\)](#), [Control.CreateGraphics\(\)](#), [Control.CreateHandle\(\)](#), [Control.CreateControl\(\)](#), [Control.DefWndProc\(ref Message\)](#), [Control.DestroyHandle\(\)](#), [Control.Dispose\(bool\)](#), [Control.DoDragDrop\(object, DragDropEffects\)](#), [Control.DoDragDrop\(object, DragDropEffects, Bitmap?, Point, bool\)](#), [Control.DrawToBitmap\(Bitmap, Rectangle\)](#), [Control.EndInvoke\(IAsyncResult\)](#), [Control.FindForm\(\)](#), [Control.GetTopLevel\(\)](#), [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#), [Control.RaiseMouseEvent\(object, MouseEventArgs\)](#), [Control.Focus\(\)](#),

[Control.FromChildHandle\(nint\)](#), [Control.FromHandle\(nint\)](#),
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#), [Control.GetChildAtPoint\(Point\)](#),
[Control.GetContainerControl\(\)](#), [Control.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#),
[Control.GetNextControl\(Control?, bool\)](#), [Control.GetStyle\(ControlStyles\)](#), [Control.Hide\(\)](#),
[Control.InitLayout\(\)](#), [Control.Invalidate\(Region?\)](#), [Control.Invalidate\(Region?, bool\)](#),
[Control.Invalidate\(\)](#), [Control.Invalidate\(bool\)](#), [Control.Invalidate\(Rectangle\)](#),
[Control.Invalidate\(Rectangle, bool\)](#), [Control.Invoke\(Action\)](#), [Control.Invoke\(Delegate\)](#),
[Control.Invoke\(Delegate, params object?\[\]\)](#), [Control.Invoke<T>\(Func<T>\)](#),
[Control.InvokePaint\(Control, PaintEventArgs\)](#),
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#), [Control.IsKeyLocked\(Keys\)](#),
[Control.IsInputChar\(char\)](#), [Control.IsInputKey\(Keys\)](#), [Control.IsMnemonic\(char, string?\)](#),
[Control.LogicalToDeviceUnits\(int\)](#), [Control.LogicalToDeviceUnits\(Size\)](#),
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#), [Control.NotifyInvalidate\(Rectangle\)](#),
[Control.InvokeOnClick\(Control?, EventArgs\)](#), [Control.OnAutoSizeChanged\(EventArgs\)](#),
[Control.OnBackColorChanged\(EventArgs\)](#), [Control.OnBackgroundImageChanged\(EventArgs\)](#),
[Control.OnBackgroundImageLayoutChanged\(EventArgs\)](#),
[Control.OnBindingContextChanged\(EventArgs\)](#), [Control.OnCausesValidationChanged\(EventArgs\)](#),
[Control.OnContextMenuStripChanged\(EventArgs\)](#), [Control.OnCursorChanged\(EventArgs\)](#),
[Control.OnDataContextChanged\(EventArgs\)](#), [Control.OnDockChanged\(EventArgs\)](#),
[Control.OnEnabledChanged\(EventArgs\)](#), [Control.OnFontChanged\(EventArgs\)](#),
[Control.OnForeColorChanged\(EventArgs\)](#), [Control.OnRightToLeftChanged\(EventArgs\)](#),
[Control.OnNotifyMessage\(Message\)](#), [Control.OnParentBackColorChanged\(EventArgs\)](#),
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#),
[Control.OnParentBindingContextChanged\(EventArgs\)](#), [Control.OnParentCursorChanged\(EventArgs\)](#),
[Control.OnParentDataContextChanged\(EventArgs\)](#), [Control.OnParentEnabledChanged\(EventArgs\)](#),
[Control.OnParentFontChanged\(EventArgs\)](#), [Control.OnParentForeColorChanged\(EventArgs\)](#),
[Control.OnParentRightToLeftChanged\(EventArgs\)](#), [Control.OnParentVisibleChanged\(EventArgs\)](#),
[Control.OnPrint\(PaintEventArgs\)](#), [Control.OnTabIndexChanged\(EventArgs\)](#),
[Control.OnTabStopChanged\(EventArgs\)](#), [Control.OnTextChanged\(EventArgs\)](#),
[Control.OnVisibleChanged\(EventArgs\)](#), [Control.OnParentChanged\(EventArgs\)](#),
[Control.OnClick\(EventArgs\)](#), [Control.OnClientSizeChanged\(EventArgs\)](#),
[Control.OnControlAdded\(ControlEventArgs\)](#), [Control.OnControlRemoved\(ControlEventArgs\)](#),
[Control.OnCreateControl\(\)](#), [Control.OnHandleCreated\(EventArgs\)](#),
[Control.OnLocationChanged\(EventArgs\)](#), [Control.OnHandleDestroyed\(EventArgs\)](#),
[Control.OnDoubleClick\(EventArgs\)](#), [Control.OnDragEnter\(DragEventArgs\)](#),
[Control.OnDragOver\(DragEventArgs\)](#), [Control.OnDragLeave\(EventArgs\)](#),
[Control.OnDragDrop\(DragEventArgs\)](#), [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#),
[Control.OnEnter\(EventArgs\)](#), [Control.InvokeGotFocus\(Control?, EventArgs\)](#),
[Control.OnGotFocus\(EventArgs\)](#), [Control.OnHelpRequested\(HelpEventArgs\)](#),
[Control.OnInvalidated\(InvalidateEventArgs\)](#), [Control.OnKeyDown\(KeyEventArgs\)](#),

[Control.OnKeyPress\(KeyPressEventArgs\)](#), [Control.OnKeyUp\(KeyEventArgs\)](#),
[Control.OnLayout\(LayoutEventArgs\)](#), [Control.OnLeave\(EventArgs\)](#),
[Control.InvokeLostFocus\(Control?, EventArgs\)](#), [Control.OnLostFocus\(EventArgs\)](#),
[Control.OnMarginChanged\(EventArgs\)](#), [Control.OnMouseDoubleClick\(MouseEventArgs\)](#),
[Control.OnMouseClicked\(MouseEventArgs\)](#), [Control.OnMouseCaptureChanged\(EventArgs\)](#),
[Control.OnMouseDown\(MouseEventArgs\)](#), [Control.OnMouseEnter\(EventArgs\)](#),
[Control.OnMouseLeave\(EventArgs\)](#), [Control.OnDpiChangedBeforeParent\(EventArgs\)](#),
[Control.OnDpiChangedAfterParent\(EventArgs\)](#), [Control.OnMouseHover\(EventArgs\)](#),
[Control.OnMouseMove\(MouseEventArgs\)](#), [Control.OnMouseUp\(MouseEventArgs\)](#),
[Control.OnMouseWheel\(MouseEventArgs\)](#), [Control.OnMove\(EventArgs\)](#),
[Control.OnPaint\(PaintEventArgs\)](#), [Control.OnPaddingChanged\(EventArgs\)](#),
[Control.OnPaintBackground\(PaintEventArgs\)](#),
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#),
[Control.OnRegionChanged\(EventArgs\)](#), [Control.OnResize\(EventArgs\)](#),
[Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#), [Control.OnSizeChanged\(EventArgs\)](#),
[Control.OnChangeUICues\(UICuesEventArgs\)](#), [Control.OnStyleChanged\(EventArgs\)](#),
[Control.OnSystemColorsChanged\(EventArgs\)](#), [Control.OnValidating\(CancelEventArgs\)](#),
[Control.OnValidated\(EventArgs\)](#), [Control.RescaleConstantsForDpi\(int, int\)](#),
[Control.PerformLayout\(\)](#), [Control.PerformLayout\(Control?, string?\)](#), [Control.PointToClient\(Point\)](#),
[Control.PointToScreen\(Point\)](#), [Control.PreProcessMessage\(ref Message\)](#),
[Control.PreProcessControlMessage\(ref Message\)](#), [Control.ProcessCmdKey\(ref Message, Keys\)](#),
[Control.ProcessDialogChar\(char\)](#), [Control.ProcessDialogKey\(Keys\)](#),
[Control.ProcessKeyEventArgs\(ref Message\)](#), [Control.ProcessKeyMessage\(ref Message\)](#),
[Control.ProcessKeyPreview\(ref Message\)](#), [Control.ProcessMnemonic\(char\)](#),
[Control.RaiseDragEvent\(object, DragEventArgs\)](#), [Control.RaisePaintEvent\(object, PaintEventArgs\)](#),
[Control.RecreateHandle\(\)](#), [Control.RectangleToClient\(Rectangle\)](#),
[Control.RectangleToScreen\(Rectangle\)](#), [Control.ReflectMessage\(nint, ref Message\)](#),
[Control.Refresh\(\)](#), [Control.ResetMouseEventArgs\(\)](#), [Control.ResetText\(\)](#), [Control.ResumeLayout\(\)](#),
[Control.ResumeLayout\(bool\)](#), [Control.Scale\(SizeF\)](#), [Control.ScaleControl\(SizeF, BoundsSpecified\)](#),
[Control.Select\(\)](#), [Control.Select\(bool, bool\)](#),
[Control.SelectNextControl\(Control?, bool, bool, bool, bool\)](#), [Control.SendToBack\(\)](#),
[Control.SetBounds\(int, int, int, int\)](#), [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#),
[Control.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#), [Control.SetClientSizeCore\(int, int\)](#),
[Control.SizeFromClientSize\(Size\)](#), [Control.SetStyle\(ControlStyles, bool\)](#), [Control.SetTopLevel\(bool\)](#),
[Control.SetVisibleCore\(bool\)](#), [Control.RtlTranslateAlignment\(HorizontalAlignment\)](#),
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#),
[Control.RtlTranslateAlignment\(ContentAlignment\)](#),
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#),
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#), [Control.RtlTranslateContent\(ContentAlignment\)](#),
[Control.Show\(\)](#), [Control.SuspendLayout\(\)](#), [Control.Update\(\)](#), [Control.UpdateBounds\(\)](#),

[Control.UpdateBounds\(int, int, int, int\)](#), [Control.UpdateBounds\(int, int, int, int, int, int\)](#), [Control.UpdateZOrder\(\)](#), [Control.UpdateStyles\(\)](#), [Control.WndProc\(ref Message\)](#), [Control.OnImeModeChanged\(EventArgs\)](#), [Control.AccessibilityObject](#), [Control.AccessibleDefaultActionDescription](#), [Control.AccessibleDescription](#), [Control.AccessibleName](#), [Control.AccessibleRole](#), [Control.AllowDrop](#), [Control.Anchor](#), [Control.AutoScrollOffset](#), [Control.LayoutEngine](#), [Control.DataContext](#), [Control.BackColor](#), [Control.BackgroundImage](#), [Control.BackgroundImageLayout](#), [Control.BindingContext](#), [Control.Bottom](#), [Control.Bounds](#), [Control.CanFocus](#), [Control.CanRaiseEvents](#), [Control.CanSelect](#), [Control.Capture](#), [Control.CausesValidation](#), [Control.CheckForIllegalCrossThreadCalls](#), [Control.ClientRectangle](#), [Control.ClientSize](#), [Control.CompanyName](#), [Control.ContainsFocus](#), [Control.ContextMenuStrip](#), [Control.Controls](#), [Control.Created](#), [Control.CreateParams](#), [Control.Cursor](#), [Control.DataBindings](#), [Control.DefaultBackColor](#), [Control.DefaultCursor](#), [Control.DefaultFont](#), [Control.DefaultForeColor](#), [Control.DefaultMargin](#), [Control.DefaultMaximumSize](#), [Control.DefaultMinimumSize](#), [Control.DefaultPadding](#), [Control.DefaultSize](#), [Control.DeviceDpi](#), [Control.DisplayRectangle](#), [Control.IsDisposed](#), [Control.Disposing](#), [Control.Dock](#), [Control.DoubleBuffered](#), [Control.Enabled](#), [Control.Focused](#), [Control.Font](#), [Control.FontHeight](#), [Control.ForeColor](#), [Control.Handle](#), [Control.HasChildren](#), [Control.Height](#), [Control.IsHandleCreated](#), [Control.InvokeRequired](#), [Control.IsAccessible](#), [Control.IsAncestorSiteInDesignMode](#), [Control.IsMirrored](#), [Control.Left](#), [Control.Location](#), [Control.Margin](#), [Control.MaximumSize](#), [Control.MinimumSize](#), [Control.ModifierKeys](#), [Control.MouseButtons](#), [Control.MousePosition](#), [Control.Name](#), [Control.Parent](#), [Control.ProductName](#), [Control.ProductVersion](#), [Control.RecreatingHandle](#), [Control.Region](#), [Control.RenderRightToLeft](#), [Control.ResizeRedraw](#), [Control.Right](#), [Control.RightToLeft](#), [Control.ScaleChildren](#), [Control.Site](#), [Control.Size](#), [Control.TabIndex](#), [Control.TabStop](#), [Control.Tag](#), [Control.Text](#), [Control.Top](#), [Control.TopLevelControl](#), [Control.ShowKeyboardCues](#), [Control.ShowFocusCues](#), [Control.UseWaitCursor](#), [Control.Visible](#), [Control.Width](#), [Control.PreferredSize](#), [Control.Padding](#), [Control.CanEnableIme](#), [Control.DefaultImeMode](#), [Control.ImeMode](#), [Control.ImeModeBase](#), [Control.PropagatingImeMode](#), [Control.BackColorChanged](#), [Control.BackgroundImageChanged](#), [Control.BackgroundImageLayoutChanged](#), [Control.BindingContextChanged](#), [Control.CausesValidationChanged](#), [Control.ClientSizeChanged](#), [Control.ContextMenuStripChanged](#), [Control.CursorChanged](#), [Control.DockChanged](#), [Control.EnabledChanged](#), [Control.FontChanged](#), [Control.ForeColorChanged](#), [Control.LocationChanged](#), [Control.MarginChanged](#), [Control.RegionChanged](#), [Control.RightToLeftChanged](#), [Control.SizeChanged](#), [Control.TabIndexChanged](#), [Control.TabStopChanged](#), [Control.TextChanged](#), [Control.VisibleChanged](#), [Control.Click](#), [Control.ControlAdded](#), [Control.ControlRemoved](#), [Control.DataContextChanged](#), [Control.DragDrop](#), [Control.DragEnter](#), [Control.DragOver](#), [Control.DragLeave](#), [Control.GiveFeedback](#), [Control.HandleCreated](#), [Control.HandleDestroyed](#),

[Control.HelpRequested](#) , [Control.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) , [Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) , [Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) , [Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) , [Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) , [Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) , [Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) , [Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) , [Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) , [Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) , [Component.Dispose\(\)](#) , [Component.Dispose\(bool\)](#) , [Component.GetService\(Type\)](#) , [Component.ToString\(\)](#) , [Component.CanRaiseEvents](#) , [Component.Container](#) , [Component.DesignMode](#) , [Component.Events](#) , [Component.Site](#) , [Component.Disposed](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#)

Constructors

Form1()

Initializes a new instance of the ASE_Assignment.Form1 class, setting up the canvas, command factory, stored program, and parser.

```
public Form1()
```

Methods

Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

disposing [bool](#)

true if managed resources should be disposed; otherwise, false.

Namespace BOOSEprojectTests

Classes

[AppCanvasTests](#)

unit test class for testing the command functions of ASE_Assignment.AppCanvas

[AppCircleTests](#)

Unit test class for testing the functionality of the ASE_Assignment.AppCircle command.

[AppCommandFactoryTests](#)

Unit test class for testing the ASE_Assignment.AppCommandFactory class.

[AppDrawToTests](#)

Unit test class for testing the functionality of the ASE_Assignment.AppDrawTo command.

[AppMoveToTests](#)

Unit test class for testing the functionality of the ASE_Assignment.AppMoveTo command.

[AppPenColourTests](#)

Unit tests for the ASE_Assignment.AppPenColour command.

[AppRectTests](#)

Unit test class for testing the functionality of the ASE_Assignment.AppRect command.

Class AppCanvasTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








unit test class for testing the command functions of ASE_Assignment.AppCanvas

```
[TestClass]  
public class AppCanvasTests
```

Inheritance

[object](#)  ← [AppCanvasTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

DrawToStoresPenPositionCorrectly()

Tests if the ASE_Assignment.AppCanvas.DrawTo(System.Int32,System.Int32) method correctly stores the pen position inputted.

```
[TestMethod]  
public void DrawToStoresPenPositionCorrectly()
```

MoveToStoresPenPositionCorrectly()

Tests if the ASE_Assignment.AppCanvas.MoveTo(System.Int32,System.Int32) method correctly stores pen position inputted.

```
[TestMethod]  
public void MoveToStoresPenPositionCorrectly()
```

MultilineInputExecutesAllCommandsCorrectly()

Tests to see if pen position is still stored correctly after multiple inputs of ASE_Assignment.AppCanvas.DrawTo(System.Int32, System.Int32) and ASE_Assignment.AppCanvas.MoveTo(System.Int32, System.Int32) commands in ASE_Assignment.AppCanvas.

```
[TestMethod]
```

```
public void MultilineInputExecutesAllCommandsCorrectly()
```


Class AppCircleTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit test class for testing the functionality of the ASE_Assignment.AppCircle command.

```
[TestClass]
public class AppCircleTests
```

Inheritance

[object](#)  ← [AppCircleTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

Execute_AppCircleCommandDrawsCorrectly()

Tests if the ASE_Assignment.AppCircle.Execute method correctly draws a circle on the canvas with the given radius and filled status.

```
[TestMethod]
public void Execute_AppCircleCommandDrawsCorrectly()
```

Remarks

Test arranged by providing the method with valid parameters. After ASE_Assignment.AppCircle command is executed, test checks if the X and Y positions of canvas remain the same as before the command was executed. If they remain unchanged the test is successful.

Execute_AppCircleCommandThrowsExceptionForInvalidParameters()

Tests if the ASE_Assignment.AppCircle.Execute method throws a System.FormatException when the radius parameter cannot be parsed as an integer.

```
[TestMethod]
[ExpectedException(typeof(FormatException))]
public void Execute_AppCircleCommandThrowsExceptionForInvalidParameters()
```

Remarks

Test arranged by attempting to input a string value as the radius parameter which is invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If FormatException is thrown, test will be successful. Any other outcome results in a failed test.

Execute_AppCircleCommandThrowsExceptionForMissingParameters()

Tests if the ASE_Assignment.AppCircle.Execute method throws an System.IndexOutOfRangeException when the command is provided with insufficient parameters.

```
[TestMethod]
[ExpectedException(typeof(IndexOutOfRangeException))]
public void Execute_AppCircleCommandThrowsExceptionForMissingParameters()
```

Remarks

Test arranged by attempting to execute ASE_Assignment.AppCircle command without providing it with a parameter. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If IndexOutOfRangeException is thrown, test will be successful. Any other outcome results in a failed test.

Execute_AppCircleCommandThrowsExceptionForNegativeRadius()

Tests if the ASE_Assignment.AppCircle.Execute method throws an BOOSE.CanvasException when the radius parameter is invalid.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
```

```
public void Execute_AppCircleCommandThrowsExceptionForNegativeRadius()
```

Remarks

Test arranged by setting canvas and attempting to input a negative value as the radius parameter, which would make it invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If CanvasException is thrown, test will be successful. Any other outcome results in a failed test.

Class AppCommandFactoryTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit test class for testing the ASE_Assignment.AppCommandFactory class.

```
[TestClass]
public class AppCommandFactoryTests
```

Inheritance

[object](#)  ← [AppCommandFactoryTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

MakeCommand_ReturnsAppCircle_WhenCircleCommandIsPassed()

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method creates the ASE_Assignment.AppCircle command correctly.

```
[TestMethod]
public void MakeCommand_ReturnsAppCircle_WhenCircleCommandIsPassed()
```

MakeCommand_ReturnsAppDrawTo_WhenDrawToCommandIsPassed()

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method creates the ASE_Assignment.AppDrawTo command correctly.

```
[TestMethod]
public void MakeCommand_ReturnsAppDrawTo_WhenDrawToCommandIsPassed()
```

MakeCommand_ReturnsAppPenColour_WhenPenColourCommandIsPassed()

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method creates the ASE_Assignment.AppPenColour command correctly.

```
[TestMethod]
public void MakeCommand_ReturnsAppPenColour_WhenPenColourCommandIsPassed()
```

MakeCommand_ReturnsAppRect_WhenRectCommandIsPassed()

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method creates the ASE_Assignment.AppRect command correctly.

```
[TestMethod]
public void MakeCommand_ReturnsAppRect_WhenRectCommandIsPassed()
```

MakeCommand_ThrowsException_WhenInvalidCommandIsPassed()

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method throws a BOOSE.FactoryException when an invalid command type is passed.

```
[TestMethod]
[ExpectedException(typeof(FactoryException))]
public void MakeCommand_ThrowsException_WhenInvalidCommandIsPassed()
```

Remarks

Test is arranged by attempting to send an invalid command to the AppCommandFactory. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If FactoryException is thrown, test will be successful. Any other outcome results in a failed test.

MakeCommand_ThrowsException_WhenNullCommandIsPassed())

Tests if the ASE_Assignment.AppCommandFactory.MakeCommand(System.String) method throws a System.NullReferenceException when a null command type is passed.

```
[TestMethod]
[ExpectedException(typeof(NullReferenceException))]
public void MakeCommand_ThrowsException_WhenNullCommandIsPassed()
```

Remarks

Test is arranged by attempting to send a null command to the AppCommandFactory. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If NullReferenceException is thrown, test will be successful. Any other outcome results in a failed test.

Setup()

Initializes the ASE_Assignment.AppCommandFactory instance before each test method.

```
[TestInitialize]
public void Setup()
```

Class AppDrawToTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit test class for testing the functionality of the ASE_Assignment.AppDrawTo command.

```
[TestClass]
public class AppDrawToTests
```

Inheritance

[object](#)  ← [AppDrawToTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

Execute_AppDrawToCommandThrowsExceptionForInvalidParameters()

Tests if the ASE_Assignment.AppDrawTo.Execute method throws a System.FormatException when the command parameters cannot be parsed as integers.

```
[TestMethod]
[ExpectedException(typeof(FormatException))]
public void Execute_AppDrawToCommandThrowsExceptionForInvalidParameters()
```

Remarks

Test is arranged by setting the canvas and attempting to parse two strings as parameters which are invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is absent, the test will fail.

Execute_AppDrawToCommandThrowsExceptionForMissingParameters()

Tests if the ASE_Assignment.AppDrawTo.Execute method throws an System.IndexOutOfRangeException when the command is provided with only one parameter.

```
[TestMethod]
[ExpectedException(typeof(IndexOutOfRangeException))]
public void Execute_AppDrawToCommandThrowsExceptionForMissingParameters()
```

Remarks

Test is arranged by setting the canvas, and only providing one of the two expected parameters. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is absent, the test will fail.

Execute_AppDrawToCommandUpdatesCanvasCorrectly()

Tests if the ASE_Assignment.AppDrawTo.Execute method correctly updates the canvas position when valid parameters are provided.

```
[TestMethod]
public void Execute_AppDrawToCommandUpdatesCanvasCorrectly()
```

Remarks

Test is arranged by setting the canvas, pen colour, and 2 valid parameters for the method. After the ASE_Assignment.AppDrawTo command is executed, the X and Y coordinates are tested to ensure that the pen position on the canvas is correctly updated by the command.

Class AppMoveToTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit test class for testing the functionality of the ASE_Assignment.AppMoveTo command.

```
[TestClass]
public class AppMoveToTests
```

Inheritance

[object](#)  ← [AppMoveToTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

Execute_AppMoveToCommandThrowsExceptionForInvalidParameters()

Tests if the ASE_Assignment.AppMoveTo.Execute method throws a System.FormatException when the command parameters cannot be parsed as valid integers.

```
[TestMethod]
[ExpectedException(typeof(FormatException))]
public void Execute_AppMoveToCommandThrowsExceptionForInvalidParameters()
```

Remarks

Test is arranged by setting the canvas and attempting to parse strings as parameters which is invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is not thrown, the test will fail.

Execute_AppMoveToCommandThrowsExceptionForMissingParameters()

Tests if the ASE_Assignment.AppMoveTo.Execute method throws an System.IndexOutOfRangeException when the command is provided with insufficient parameters.

```
[TestMethod]
[ExpectedException(typeof(IndexOutOfRangeException))]
public void Execute_AppMoveToCommandThrowsExceptionForMissingParameters()
```

Remarks

Test is arranged by setting the canvas and providing only one parameter instead of the required two. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is not thrown, the test will fail.

Execute_AppMoveToCommandUpdatesCanvasCorrectly()

Tests if the ASE_Assignment.AppMoveTo.Execute method correctly updates pen position.

```
[TestMethod]
public void Execute_AppMoveToCommandUpdatesCanvasCorrectly()
```

Remarks

Test is arranged by setting the canvas and valid parameters for the method. After the ASE_Assignment.AppMoveTo command is executed, the pen positions X and Y coordinates are checked, to ensure that they have been properly updated by the command.

Class AppPenColourTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit tests for the ASE_Assignment.AppPenColour command.

```
[TestClass]
public class AppPenColourTests
```

Inheritance

[object](#)  ← [AppPenColourTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

Execute_AppPenColourCommandDoesNotChangePenColourForInvalidValues()

Verifies that the ASE_Assignment.AppPenColour command does not update the pen color on the canvas when invalid RGB values are provided.

```
[TestMethod]
public void Execute_AppPenColourCommandDoesNotChangePenColourForInvalidValues()
```

Remarks

Test attempts to execute ASE_Assignment.AppPenColour command with R:G:B values above 255, which makes the parameters invalid.

Execute_AppPenColourCommandThrowsExceptionForInvalidParameters()

Tests that the ASE_Assignment.AppPenColour command throws a System.FormatException when invalid parameters are provided.

```
[TestMethod]
[ExpectedException(typeof(FormatException))]
public void Execute_AppPenColourCommandThrowsExceptionForInvalidParameters()
```

Remarks

Test is arranged by attempting to parse strings as parameters which is invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If FormatException is thrown, test will be successful. Any other outcome results in a failed test.

Execute_AppPenColourCommandThrowsExceptionForMissingParameters()

Verifies that the ASE_Assignment.AppPenColour command throws an System.IndexOutOfRangeException when insufficient parameters are provided.

```
[TestMethod]
[ExpectedException(typeof(IndexOutOfRangeException))]
public void Execute_AppPenColourCommandThrowsExceptionForMissingParameters()
```

Remarks

Test attempts to execute the ASE_Assignment.AppPenColour command without assigning a blue value. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If IndexOutOfRangeException is thrown, test will be successful. Any other outcome results in a failed test.

Execute_AppPenColourCommandUpdatesPenColourCorrectly()

Tests that the ASE_Assignment.AppPenColour command correctly updates the pen color on the canvas.

```
[TestMethod]
public void Execute_AppPenColourCommandUpdatesPenColourCorrectly()
```

Setup()

Initializes instances of ASE_Assignment.AppCanvas and ASE_Assignment.AppPenColour before each test.

```
[TestInitialize]  
public void Setup()
```

Class AppRectTests

Namespace: [BOOSEprojectTests](#)

Assembly: BOOSEprojectTests.dll








Unit test class for testing the functionality of the ASE_Assignment.AppRect command.

```
[TestClass]
public class AppRectTests
```

Inheritance

[object](#)  ← [AppRectTests](#)

Inherited Members

[object.Equals\(object?\)](#) , [object.Equals\(object?, object?\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object?, object?\)](#) , [object.ToString\(\)](#) 

Methods

Execute_AppRectCommandThrowsExceptionForInvalidParameters()

Tests if the ASE_Assignment.AppRect.Execute method throws a System.FormatException when the command parameters cannot be parsed as valid integers.

```
[TestMethod]
[ExpectedException(typeof(FormatException))]
public void Execute_AppRectCommandThrowsExceptionForInvalidParameters()
```

Remarks

Test is arranged by setting the canvas and attempting to parse strings as parameters which is invalid. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is not thrown, the test will fail.

Execute_AppRectCommandThrowsExceptionForMissingParameter()

Tests if the ASE_Assignment.AppRect.Execute method throws an System.IndexOutOfRangeException when the command is provided with insufficient parameters.

```
[TestMethod]
[ExpectedException(typeof(ArgumentNullException))]
public void Execute_AppRectCommandThrowsExceptionForMissingParameter()
```

Remarks

Test is arranged by setting the canvas and providing the method with the width and height parameters, but no filled parameter. Test does not require an Assert, as the [ExpectedException] attribute acts as the assertion itself. If the expected exception is not thrown, the test will fail.

Execute_AppRectCommandUpdatesCanvasCorrectly()

Tests if the ASE_Assignment.AppRect.Execute method correctly draws the rectangle on the canvas without altering pen position.

```
[TestMethod]
public void Execute_AppRectCommandUpdatesCanvasCorrectly()
```

Remarks

Test is arranged by setting the canvas, pen colour, and valid parameters for the method. The pen position is then set to a specific value to measure whether or not the execution of the ASE_Assignment.AppRect command alters the position of the pen. After the ASE_Assignment.AppRect command is executed, the X and Y coordinates are tested to ensure that the pen position on the canvas is correctly updated by the command.