

# Namespace ASE\_Assignment

## Classes

### [Form1](#)

The main form for the drawing application, responsible for UI interactions, canvas operations, and program execution.

# Class Form1

Namespace: [ASE Assignment](#)

Assembly: ASE\_Assignment.dll

The main form for the drawing application, responsible for UI interactions, canvas operations, and program execution.

```
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.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.ScaleMinAxisSize\(float, float, bool\)](#) ,  
[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.OnAutoValidateChanged\(EventArgs\)](#) ,  
[ContainerControl.OnMove\(EventArgs\)](#) , [ContainerControl.OnParentChanged\(EventArgs\)](#) ,  
[ContainerControl.PerformLayout\(\)](#) , [ContainerControl.RescaleConstantsForDpi\(int, int\)](#) ,  
[ContainerControl/Validate\(\)](#) , [ContainerControl/Validate\(bool\)](#) ,  
[ContainerControl/AutoScaleDimensions](#) , [ContainerControl/AutoScaleFactor](#) ,  
[ContainerControl/AutoScaleMode](#) , [ContainerControl/BindingContext](#) ,  
[ContainerControl/CanEnableIme](#) , [ContainerControl/ActiveControl](#) ,

[ContainerControl.CurrentAutoScaleDimensions](#) , [ContainerControl.ParentForm](#) ,  
[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,  
[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,  
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,  
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,  
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,  
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,  
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,  
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[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.CreateGraphics\(\)](#) ,  
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [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.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.IsAnyInputChar\(char\)](#) , [Control.IsAnyInputKey\(Keys\)](#) , [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.OnBindingContextChanged\(EventArgs\)](#) ,  
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,

[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,  
[Control.OnDockChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,  
[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.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnLocationChanged\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.InvokeGotFocus\(Control, EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidate\(InvalidateEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyPressEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[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.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnSystemColorsChanged\(EventArgs\)](#) , [Control.OnValidating\(CancelEventArgs\)](#) ,  
[Control.OnValidated\(EventArgs\)](#) , [Control.PerformLayout\(\)](#) , [Control.PerformLayout\(Control, string\)](#) ,  
[Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[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.Select\(\)](#) ,  
[Control.SelectNextControl\(Control, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,

[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,  
[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.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.DataContext](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.CompanyName](#) ,  
[Control.ContainsFocus](#) , [Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) ,  
[Control.Cursor](#) , [Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) ,  
[Control.DefaultFont](#) , [Control.DefaultForeColor](#) , [Control.DefaultMargin](#) ,  
[Control.DefaultMaximumSize](#) , [Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) ,  
[Control.DeviceDpi](#) , [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.Accessible](#) ,  
[Control.IsAncestorSiteInDesignMode](#) , [Control.IsMirrored](#) , [Control.Left](#) , [Control.Margin](#) ,  
[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.TabIndex](#) , [Control.TabStop](#) ,  
[Control.Tag](#) , [Control.Top](#) , [Control.TopLevelControl](#) , [Control.ShowKeyboardCues](#) ,  
[Control.ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) , [Control.Width](#) ,  
[Control.PreferredSize](#) , [Control.Padding](#) , [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.Invalidate](#) , [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.ChangeUICTypes](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,  
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,  
[Component.Dispose\(\)](#) , [Component.GetService\(Type\)](#) , [Component.Container](#) ,  
[Component.DesignMode](#) , [Component.Events](#) , [Component.Disposed](#) ,  
[MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) ,  
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

## Constructors

### Form1()

Constructor to initializes the form and sets up required components.

```
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 ASE\_Assignment.Builder\_Design\_Pattern

## Classes

### [AppComponentsBuilder](#)

Builder class for constructing components including canvas, command factory, stored program, and parser. Implements the [IAppComponentsBuilder](#) interface.

### [AppComponentsDirector](#)

Directs the construction process of application components using the builder pattern.

## Interfaces

### [IAppComponentsBuilder](#)

Interface to defines the contract for building components

# Class AppComponentsBuilder

Namespace: [ASE Assignment.Builder Design Pattern](#)

Assembly: ASE\_Assignment.dll

Builder class for constructing components including canvas, command factory, stored program, and parser. Implements the [IAppComponentsBuilder](#) interface.

```
public class AppComponentsBuilder : IAppComponentsBuilder
```

**Inheritance**

[object](#) ← AppComponentsBuilder

**Implements**

[IAppComponentsBuilder](#)

**Inherited Members**

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

## Constructors

### AppComponentsBuilder()

Constructor to initializes a new instance of the [AppComponentsBuilder](#) class.

```
public AppComponentsBuilder()
```

## Methods

### BuildCanvas()

Builds a new instance of the canvas component.

```
public void BuildCanvas()
```

## BuildCommandFactory()

Builds a new instance of the command factory component.

```
public void BuildCommandFactory()
```

## BuildParser()

Builds a new instance of the parser component.

```
public void BuildParser()
```

## BuildStoredProgram()

Builds a new instance of the stored program component.

```
public void BuildStoredProgram()
```

## GetCanvas()

Gets the constructed canvas component.

```
public AppCanvas GetCanvas()
```

Returns

[AppCanvas](#)

An instance of [AppCanvas](#).

## GetCommandFactory()

Gets the constructed command factory component.

```
public AppCommandFactory GetCommandFactory()
```

Returns

[AppCommandFactory](#)

An instance of [AppCommandFactory](#).

## GetParser()

Gets the constructed parser component.

```
public AppParser GetParser()
```

Returns

[AppParser](#)

An instance of [AppParser](#).

## GetStoredProgram()

Gets the constructed stored program component.

```
public AppStoredProgram GetStoredProgram()
```

Returns

[AppStoredProgram](#)

An instance of [AppStoredProgram](#).

# Class AppComponentsDirector

Namespace: [ASE Assignment.Builder Design Pattern](#)

Assembly: ASE\_Assignment.dll

Directs the construction process of application components using the builder pattern.

```
public class AppComponentsDirector
```

## Inheritance

[object](#) ← AppComponentsDirector

## Inherited Members

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

## Constructors

### AppComponentsDirector(IAppComponentsBuilder)

Construtor to initializes a new instance of the [AppComponentsDirector](#) class with the specified components builder.

```
public AppComponentsDirector(IAppComponentsBuilder componentsBuilder)
```

## Parameters

`componentsBuilder` [IAppComponentsBuilder](#)

The builder responsible for constructing application components.

## Methods

### Build()

Builds all necessary components in a predefined order.

```
public void Build()
```

# Interface IAppComponentsBuilder

Namespace: [ASE Assignment.Builder Design Pattern](#)

Assembly: ASE\_Assignment.dll

Interface to defines the contract for building components

```
public interface IAppComponentsBuilder
```

## Methods

### BuildCanvas()

Builds the canvas component

```
void BuildCanvas()
```

### BuildCommandFactory()

Builds the command factory component

```
void BuildCommandFactory()
```

### BuildParser()

Builds the parser component

```
void BuildParser()
```

### BuildStoredProgram()

Builds the stored program component

```
void BuildStoredProgram()
```

## GetCanvas()

Gets the constructed canvas component.

```
AppCanvas GetCanvas()
```

Returns

[AppCanvas](#)

An instance of [AppCanvas](#).

## GetCommandFactory()

Gets the constructed command factory component.

```
AppCommandFactory GetCommandFactory()
```

Returns

[AppCommandFactory](#)

An instance of [AppCommandFactory](#).

## GetParser()

Gets the constructed parser component.

```
AppParser GetParser()
```

Returns

[AppParser](#)

An instance of [AppParser](#).

## GetStoredProgram()

Gets the constructed stored program component.

AppStoredProgram **GetStoredProgram()**

Returns

[AppStoredProgram](#)

An instance of [AppStoredProgram](#).

# Namespace ASE\_Assignment.Commands

## Classes

### [AppArray](#)

Constructor to initializes a new instance of the [AppArray](#) class. Reduces the restriction counter in the BOOSE.Array class when initialized.

### [AppCircle](#)

Command to draw a circle on the canvas. Allows drawing a circle with specified radius and an option to fill flag.

### [AppElse](#)

Else implementation that inherits functionality from the BOOSE.Else class.

### [AppEnd](#)

End implementation that inherits functionality from the BOOSE.End class.

### [AppFor](#)

For implementation that inherits functionality from the BOOSE.For class.

### [AppIf](#)

If implementation that inherits functionality from the BOOSE.If class.

### [AppInt](#)

Int implementation that inherits functionality from the BOOSE.Int class.

### [AppMethod](#)

Method implementation that inherits functionality from the BOOSE.Method class.

### [AppReal](#)

Real implementation that inherits functionality from the BOOSE.Real class.

### [AppRectangle](#)

Command to draw a rectangle on the canvas. Allows drawing a rectangle with specified dimensions and an option to fill flag.

### [AppStar](#)

Command to draw a star on the canvas. Allows drawing a star with specified dimensions and an option to fill flag.

### [AppTriangle](#)

Command to draw a triangle on the canvas. Inherits from CommandTwoParameters to utilize its structure for parameter handling.

## [AppWhile](#)

While implementation that inherits functionality from the BOOSE.While class.

## [AppWrite](#)

Command to write text on the canvas. Inherits from CommandOneParameters to utilize its structure for parameter handling.

# Class AppArray

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Constructor to initializes a new instance of the [AppArray](#) class. Reduces the restriction counter in the BOOSE.Array class when initialized.

```
public class AppArray : Array, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Array ← AppArray

## Implements

ICommand

## Inherited Members

Array.PEEK , Array.POKE , Array.type , Array.rows , Array.columns , Array.valueInt , Array.valueReal ,  
Array.intArray , Array.realArray , Array.pokeValue , Array.peekVar , Array.rowS , Array.columnS , Array.row ,  
Array.column , Array.ArrayRestrictions() , Array.ReduceRestrictionCounter() , Array.Compile() ,  
[Array.CheckParameters\(string\[\]\)](#) , Array.Execute() , [Array.ProcessArrayParametersCompile\(bool\)](#) ,  
[Array.ProcessArrayParametersExecute\(bool\)](#) , [Array.SetIntArray\(int, int, int\)](#) ,  
[Array.SetRealArray\(double, int, int\)](#) , [Array.GetIntArray\(int, int\)](#) , [Array.GetRealArray\(int, int\)](#) ,  
Array.Rows , Array.Columns , Evaluation.expression , Evaluation.evaluatedExpression ,  
Evaluation.varName , Evaluation.value , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,  
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,  
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,  
[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\(object, object\)](#)

## Constructors

### AppArray()

Constructor to initializes a new instance of the [AppArray](#) class. Reduces the restriction counter in the BOOSE.Array class when initialized.

```
public AppArray()
```

# Class AppCircle

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Command to draw a circle on the canvas. Allows drawing a circle with specified radius and an option to fill flag.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [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\(object, object\)](#)

## Constructors

### AppCircle()

Constructor to initializes the AppCircle and sets up required components.

```
public AppCircle()
```

## AppCircle(Canvas, int, string)

Initializes a new instance of the [AppCircle](#) class with specified parameters.

```
public AppCircle(Canvas c, int radius, string fill)
```

### Parameters

c Canvas

The canvas where the circle will be drawn.

radius [int](#)

The radius of the circle.

fill [string](#)

Specifies whether the circle should be filled.

## Methods

### CheckParameters(string[])

Validates the parameters for the circle command. Ensures the parameter count is either 1 or 2.

```
public override void CheckParameters(string[] parameterList)
```

### Parameters

parameterList [string](#)[]

The array of parameters to validate.

### Exceptions

CommandException

Thrown when the number of parameters is less than 1 or greater than 2.

## Execute()

Executes the circle command by drawing a circle on the canvas.

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

## Exceptions

RestrictionException

Thrown when the radius exceeds allowed limits.

# Class AppElse

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Else implementation that inherits functionality from the BOOSE.Else class.

```
public class AppElse : Else, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Else ← AppElse

## Implements

ICommand

## Inherited Members

[Else.CheckParameters\(string\[\]\)](#) , Else.Compile() , Else.Execute() , Else.CorrespondingEnd ,  
CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrespondingCommand ,  
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,  
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,  
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,  
Command.Parameters , Command.Parmsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

## Constructors

### AppElse()

Constructor to initializes a new instance of the [AppElse](#) class. Reduces the restriction in the BOOSE.CompoundCommand class when initialized.

```
public AppElse()
```

## Methods

### Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppEnd

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

End implementation that inherits functionality from the BOOSE.End class.

```
public class AppEnd : End, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← End ← AppEnd

## Implements

ICommand

## Inherited Members

End.Compile() , CompoundCommand.ReduceRestrictions() ,  
[CompoundCommand.CheckParameters\(string\[\]\)](#) , CompoundCommand.CorrespondingCommand ,  
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,  
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,  
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,  
Command.Parameters , Command.Parmsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

## Constructors

### AppEnd()

Constructor to initializes a new instance of the [AppEnd](#) class. Reduces the restriction in the BOOSE.CompoundCommand class when initialized.

```
public AppEnd()
```

## Methods

### Execute()

Overrides the BOOSE.End.Execute() method to remove the restrictions. It handles specific logic for **While**, **For**, and **Method** commands.

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

### Exceptions

#### CommandException

Thrown when a variable does not exist or when invalid loop conditions are encountered.

### Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppFor

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

For implementation that inherits functionality from the BOOSE.For class.

```
public class AppFor : For, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← For ← AppFor

## Implements

ICommand

## Inherited Members

For.Compile() , For.Execute() , For.LoopControlV , For.From , For.To , For.Step ,  
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,  
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,  
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,  
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,  
[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\(object, object\)](#)

## Constructors

### AppFor()

Constructor to initializes a new instance of the [AppFor](#) class.

```
public AppFor()
```

# Methods

## Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppIf

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

End implementation that inherits functionality from the BOOSE.If class.

```
public class AppIf : If, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← If ← AppIf

## Implements

ICommand

## Inherited Members

CompoundCommand.ReduceRestrictions() , [CompoundCommand.CheckParameters\(string\[\]\)](#) ,  
CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand ,  
ConditionalCommand.endLineNumber , ConditionalCommand.Execute() ,  
ConditionalCommand.EndLineNumber , ConditionalCommand.Condition ,  
ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [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\(object, object\)](#)

## Constructors

### AppIf()

Constructor to initializes a new instance of the [AppIf](#) class. Reduces the restriction in the BOOSE.CompoundCommand class when initialized.

```
public AppIf()
```

## Methods

### Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppInt

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Int implementation that inherits functionality from the BOOSE.Int class.

```
public class AppInt : Int, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Int ← AppInt

## Implements

ICommand

## Inherited Members

Int.Compile() , Int.Execute() , Evaluation.expression , Evaluation.evaluatedExpression ,  
Evaluation.varName , Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [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\(object, object\)](#)

# Constructors

## AppInt()

Constructor to initializes a new instance of the [AppInt](#) class.

```
public AppInt()
```

# Methods

## Restrictions()

Overrides the BOOSE.Int.Restrictions() method.

```
public override void Restrictions()
```

# Class AppMethod

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Method implementation that inherits functionality from the BOOSE.Method class.

```
public class AppMethod : Method, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Method ← AppMethod

## Implements

ICommand

## Inherited Members

[Method.CheckParameters\(string\[\]\)](#) , Method.Compile() , Method.Execute() , Method.LocalVariables ,  
Method.MethodName , Method.Type , CompoundCommand.ReduceRestrictions() ,  
CompoundCommand.CorrectingCommand , ConditionalCommand.endLineNumber ,  
ConditionalCommand.EndLineNumber , ConditionalCommand.Condition ,  
ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [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\(object, object\)](#)

## Constructors

### AppMethod()

Constructor to initializes a new instance of the [AppMethod](#) class.

Reduces the restriction in the BOOSE.CompoundCommand class when initialized.

```
public AppMethod()
```

## Methods

### Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppReal

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Int implementation that inherits functionality from the BOOSE.Real class.

```
public class AppReal : Real, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Real ← AppReal

## Implements

ICommand

## Inherited Members

Real.Compile() , Real.Execute() , Real.Value , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [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\(object, object\)](#)

## Constructors

### AppReal()

Constructor to initializes a new instance of the [AppReal](#) class.

```
public AppReal()
```

## Methods

### Restrictions()

Overrides the BOOSE.Real.Restrictions() method.

```
public override void Restrictions()
```

# Class AppRectangle

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Command to draw a rectangle on the canvas. Allows drawing a rectangle with specified dimensions and an option to fill flag.

```
public class AppRectangle : CommandThreeParameters, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandThreeParameters.param3 , CommandThreeParameters.param3unprocessed ,  
CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [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\(object, object\)](#)

## Constructors

### AppRectangle()

Constructor to initializes the AppRectangle and sets up required components.

```
public AppRectangle()
```

## AppRectangle(Canvas, int, int, string)

Initializes a new instance of the [AppRectangle](#) class with specified parameters.

```
public AppRectangle(Canvas c, int length, int breadth, string fill)
```

### Parameters

c Canvas

The canvas where the rectangle will be drawn.

length int ↗

The length of the rectangle.

breadth int ↗

The breadth of the rectangle.

fill string ↗

Specifies whether the rectangle should be filled.

## Methods

### CheckParameters(string[])

Validates the parameters for the rectangle command. Ensures the parameter count is either 2 or 3.

```
public override void CheckParameters(string[] parameterList)
```

### Parameters

parameterList string[] ↗

The array of parameters to validate.

### Exceptions

CommandException

Thrown when the number of parameters is less than 2 or greater than 3.

## Execute()

Executes the rectangle command by drawing a rectangle on the canvas.

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

## Exceptions

RestrictionException

Thrown when the length or breadth exceeds allowed limits.

# Class AppStar

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Command to draw a star on the canvas. Allows drawing a star with specified dimensions and an option to fill flag.

```
public class AppStar : CanvasCommand, ICommand
```

## Inheritance

[object](#) ← Command ← CanvasCommand ← AppStar

## Implements

ICommand

## Inherited Members

CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [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\(object, object\)](#)

## Constructors

### AppStar()

Constructor to initializes a new instance of the [AppStar](#) class.

```
public AppStar()
```

## Methods

### CheckParameters(string[])

This validates the parameters for the star command.

```
public override void CheckParameters(string[] parameterList)
```

## Parameters

parameterList [string](#)[]

An array of parameterList passed to the command.

## Exceptions

CommandException

Throw CommandException when the parameterList are null, missing, or outside the allowed range.

## Execute()

This executes the star command and validates if parameters are then 1 or 2 and draws the star.

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

## Exceptions

CommandException

Throw CommandException if star parameters are missing, invalid, or exceed the allowed range.

# Class AppTriangle

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Command to draw a triangle on the canvas. Inherits from CommandTwoParameters to utilize its structure for parameter handling.

```
public class AppTriangle : CommandTwoParameters, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,  
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [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\(object, object\)](#)

## Constructors

### AppTriangle()

Constructor to initializes the AppTriangle.

```
public AppTriangle()
```

## AppTriangle(Canvas, int, int)

Parameterized constructor to initialize the triangle with a canvas and dimensions.

```
public AppTriangle(Canvas c, int width, int height)
```

### Parameters

c Canvas

Canvas object to draw on.

width int ↗

Width of the triangle.

height int ↗

Height of the triangle.

## Methods

### CheckParameters(string[])

Checks the parameters provided to the Tri method.

```
public override void CheckParameters(string[] parameterList)
```

### Parameters

parameterList string ↗[]

Array of parameters passed to the Tri method.

### Exceptions

CommandException

Thrown when the number of parameters is incorrect.

## Execute()

Executes the command to draw the triangle on the canvas.

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

## Exceptions

RestrictionException

Thrown if the width or height exceeds allowed limits.

# Class AppWhile

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

While implementation that inherits functionality from the BOOSE.While class.

```
public class AppWhile : While, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← While ← AppWhile

## Implements

ICommand

## Inherited Members

CompoundCommand.ReduceRestrictions() , [CompoundCommand.CheckParameters\(string\[\]\)](#) ,  
CompoundCommand.Compile() , CompoundCommand.CorrectingCommand ,  
ConditionalCommand.endLineNumber , ConditionalCommand.Execute() ,  
ConditionalCommand.EndLineNumber , ConditionalCommand.Condition ,  
ConditionalCommand.LineNumber , ConditionalCommand.CondType ,  
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,  
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [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\(object, object\)](#)

## Constructors

### AppWhile()

Constructor to initializes a new instance of the [AppWhile](#) class. Reduces the restriction in the BOOSE.CompoundCommand class when initialized.

```
public AppWhile()
```

## Methods

### Restrictions()

Overrides the BOOSE.Boolean.Restrictions() method.

```
public override void Restrictions()
```

# Class AppWrite

Namespace: [ASE Assignment.Commands](#)

Assembly: ASE\_Assignment.dll

Command to write text on the canvas. Inherits from CommandOneParameters to utilize its structure for parameter handling.

```
public class AppWrite : CommandOneParameter, ICommand
```

## Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← AppWrite

## Implements

ICommand

## Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,  
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,  
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [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\(object, object\)](#)

## Constructors

### AppWrite()

Constructor to initializes the AppWrite and sets up required components.

```
public AppWrite()
```

### AppWrite(Canvas, string)

Parameterized constructor to initialize the write command with a canvas and text.

```
public AppWrite(Canvas c, string text)
```

## Parameters

c Canvas

Canvas object to draw on.

text [string](#)

Text to write on the canvas.

## Methods

### CheckParameters(string[])

Checks the parameters provided to the Write method.

```
public override void CheckParameters(string[] parameterList)
```

## Parameters

parameterList [string](#)[]

Array of parameters passed to the Write method.

## Exceptions

CommandException

Thrown when the number of parameters is incorrect.

### Execute()

Executes the command to write text on the canvas

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

## Exceptions

### RestrictionException

Thrown if the text is null or empty.

# Namespace ASE\_Assignment.Core

## Classes

### [AppCanvas](#)

Implements a canvas application for drawing shapes, lines, and handling basic graphical operations.

### [AppCommandFactory](#)

Custom implementation of the CommandFactory for creating specific commands. Extends the base CommandFactory provided by the BOOSE framework.

### [AppParser](#)

Parses commands and programs into executable commands for the application.

### [AppStoredProgram](#)

Represents a stored program that manages the execution and storage of commands with support for conditional commands.

# Class AppCanvas

Namespace: [ASE Assignment.Core](#)

Assembly: ASE\_Assignment.dll

Implements a canvas application for drawing shapes, lines, and handling basic graphical operations.

```
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 the canvas with default size.

```
public AppCanvas()
```

# Properties

## PenColour

Getter and Setter for pen color.

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

## Property Value

## [object](#)

### XCanvasSize

Getter for \_xCanvasSize

```
public int XCanvasSize { get; }
```

Property Value

[int](#)

### Xpos

Getter and Setter for x positions.

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

Property Value

[int](#)

### YCanvasSize

Getter for \_yCanvasSize

```
public int YCanvasSize { get; }
```

Property Value

[int](#)

### Ypos

Getter and Setter for y positions.

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

Property Value

[int ↗](#)

## Methods

### Circle(int, bool)

Draws a circle at the current position.

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

Parameters

[radius \[int ↗\]\(#\)](#)

Radius of the circle.

[filled \[bool ↗\]\(#\)](#)

Whether the circle is filled.

### Clear()

Clears the canvas by filling it with gray color.

```
public void Clear()
```

### DrawTo(int, int)

Draws a line from the current position to the specified position.

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

## Parameters

x [int ↗](#)

X-coordinate of the end point.

y [int ↗](#)

Y-coordinate of the end point.

## MoveTo(int, int)

Moves the pen to the specified position without drawing.

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

## Parameters

x [int ↗](#)

X-coordinate.

y [int ↗](#)

Y-coordinate.

## Rect(int, int, bool)

Draws a rectangle at the current position.

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

## Parameters

width [int ↗](#)

Width of the rectangle.

height [int ↗](#)

Height of the rectangle.

`filled` `bool`

To define the rectangle should be filled.

## Reset()

Resets the pen position to (0, 0) and the pen color to black.

```
public void Reset()
```

## Set(int, int)

Sets the canvas size and initializes the graphics object.

```
public void Set(int xSize, int ySize)
```

### Parameters

`xSize` `int`

Width of the canvas.

`ySize` `int`

Height of the canvas.

## SetColour(int, int, int)

Sets the pen color using RGB values.

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

### Parameters

`red` `int`

Red component (0-255).

**green** [int ↗](#)

Green component (0-255).

**blue** [int ↗](#)

Blue component (0-255).

## Star(int, bool)

```
public void Star(int size, bool filled)
```

Parameters

**size** [int ↗](#)

**filled** [bool ↗](#)

## Tri(int, int)

Draws a triangle with the current position.

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

Parameters

**width** [int ↗](#)

Width of the triangle.

**height** [int ↗](#)

Height of the triangle.

## WriteText(string)

Writes text at the current position.

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

## Parameters

text [string](#)

Text to write on the canvas.

## getBitmap()

Returns the bitmap of the canvas.

```
public object getBitmap()
```

## Returns

[object](#)

Bitmap object.

# Class AppCommandFactory

Namespace: [ASE Assignment.Core](#)

Assembly: ASE\_Assignment.dll

Custom implementation of the CommandFactory for creating specific commands. Extends the base CommandFactory provided by the BOOSE framework.

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

## Inheritance

[object](#) ← CommandFactory ← AppCommandFactory

## Implements

ICommandFactory

## Inherited Members

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

## Methods

### MakeCommand(string)

Overrides the base MakeCommand method to provide support for additional commands.

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

#### Parameters

commandType [string](#)

The type of command to create, passed as a string.

#### Returns

ICommand

An instance of the appropriate ICommand implementation.



# Class AppParser

Namespace: [ASE Assignment.Core](#)

Assembly: ASE\_Assignment.dll

Parses commands and programs into executable commands for the application.

```
public class AppParser : IParser
```

Inheritance

[object](#) ← AppParser

Implements

IParser

Inherited Members

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

## Constructors

### AppParser(AppCommandFactory, AppStoredProgram)

Initializes a new instance of the [AppParser](#) class.

```
public AppParser(AppCommandFactory factory, AppStoredProgram program)
```

Parameters

**factory** [AppCommandFactory](#)

The command factory to create commands.

**program** [AppStoredProgram](#)

The stored program to manage variables and commands.

## Methods

## ParseCommand(string)

Parses a single line of text into an executable command.

```
public ICommand? ParseCommand(string Line)
```

### Parameters

Line [string](#)

The line of text to parse.

### Returns

ICommand

An instance of BOOSE.ICommand or null if the line is invalid or a comment.

## ParseProgram(string)

Parses a complete program string into executable commands.

```
public void ParseProgram(string program)
```

### Parameters

program [string](#)

The program string to parse.

### Exceptions

ParserException

Thrown if the program contains syntax errors.

# Class AppStoredProgram

Namespace: [ASE Assignment.Core](#)

Assembly: ASE\_Assignment.dll

Represents a stored program that manages the execution and storage of commands with support for conditional commands.

```
public class AppStoredProgram : StoredProgram, IList, ICollection, IEnumerable,  
ICloneable, IStoredProgram
```

## Inheritance

[object](#) ← [ArrayList](#) ← [StoredProgram](#) ← [AppStoredProgram](#)

## Implements

[IList](#), [ICollection](#), [IEnumerable](#), [ICloneable](#), [IStoredProgram](#)

## Inherited Members

[StoredProgram.SyntaxOk](#), [StoredProgram.AddMethod\(Method\)](#), [StoredProgram.GetMethod\(string\)](#),  
[StoredProgram.AddVariable\(Evaluation\)](#), [StoredProgram.GetVariable\(string\)](#),  
[StoredProgram.GetVariable\(int\)](#), [StoredProgram.FindVariable\(Evaluation\)](#),  
[StoredProgram.FindVariable\(string\)](#), [StoredProgram.VariableExists\(string\)](#),  
[StoredProgram.GetVarValue\(string\)](#), [StoredProgram.UpdateVariable\(string, int\)](#),  
[StoredProgram.UpdateVariable\(string, double\)](#), [StoredProgram.UpdateVariable\(string, bool\)](#),  
[StoredProgram.DeleteVariable\(string\)](#), [StoredProgram.IsExpression\(string\)](#),  
[StoredProgram.EvaluateExpressionWithString\(string\)](#), [StoredProgram.EvaluateExpression\(string\)](#),  
[StoredProgram.Add\(Command\)](#), [StoredProgram.NextCommand\(\)](#), [StoredProgram.ResetProgram\(\)](#),  
[StoredProgram.Commandsleft\(\)](#), [StoredProgram.PC](#), [ArrayList.Adapter\(IList\)](#), [ArrayList.Add\(object\)](#),  
[ArrayList.AddRange\(Collection\)](#), [ArrayList.BinarySearch\(int, int, object, IComparer\)](#),  
[ArrayList.BinarySearch\(object\)](#), [ArrayList.BinarySearch\(object, IComparer\)](#), [ArrayList.Clear\(\)](#),  
[ArrayList.Clone\(\)](#), [ArrayList.Contains\(object\)](#), [ArrayList.CopyTo\(Array\)](#),  
[ArrayList.CopyTo\(Array, int\)](#), [ArrayList.CopyTo\(int, Array, int, int\)](#), [ArrayList.FixedSize\(ArrayList\)](#),  
[ArrayList.FixedSize\(IList\)](#), [ArrayList.GetEnumerator\(\)](#), [ArrayList.GetEnumerator\(int, int\)](#),  
[ArrayList.GetRange\(int, int\)](#), [ArrayList.IndexOf\(object\)](#), [ArrayList.IndexOf\(object, int\)](#),  
[ArrayList.IndexOf\(object, int, int\)](#), [ArrayList.Insert\(int, object\)](#),  
[ArrayList.InsertRange\(int, Collection\)](#), [ArrayList.LastIndexOf\(object\)](#),  
[ArrayList.LastIndexOf\(object, int\)](#), [ArrayList.LastIndexOf\(object, int, int\)](#),  
[ArrayList.ReadOnly\(ArrayList\)](#), [ArrayList.ReadOnly\(IList\)](#), [ArrayList.Remove\(object\)](#),  
[ArrayList.RemoveAt\(int\)](#), [ArrayList.RemoveRange\(int, int\)](#), [ArrayList.Repeat\(object, int\)](#),  
[ArrayList.Reverse\(\)](#), [ArrayList.Reverse\(int, int\)](#), [ArrayList.SetRange\(int, Collection\)](#),

[ArrayList.Sort\(\)](#) , [ArrayList.Sort\(IComparer\)](#) , [ArrayList.Sort\(int, int, IComparer\)](#) ,  
[ArrayList.Synchronized\(ArrayList\)](#) , [ArrayList.Synchronized\(IList\)](#) , [ArrayList.ToArray\(\)](#) ,  
[ArrayList.ToArray\(Type\)](#) , [ArrayList.TrimToSize\(\)](#) , [ArrayList.Capacity](#) , [ArrayList.Count](#) ,  
[ArrayList.FixedSize](#) , [ArrayList.ReadOnly](#) , [ArrayList.Synchronized](#) , [ArrayList.this\[int\]](#) ,  
[ArrayList.SyncRoot](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### AppStoredProgram(AppCanvas)

Constructor to initializes a new instance of the [AppStoredProgram](#) class.

```
public AppStoredProgram(AppCanvas canvas)
```

#### Parameters

**canvas** [AppCanvas](#)

The canvas interface used for executing commands.

## Methods

### Pop()

Removes and returns the top BOOSE.ConditionalCommand from the stack.

```
public override ConditionalCommand Pop()
```

#### Returns

ConditionalCommand

The top conditional command.

### Exceptions

## StoredProgramException

Thrown when the stack is empty.

## Push(ConditionalCommand)

Pushes a BOOSE.ConditionalCommand onto the stack.

```
public override void Push(ConditionalCommand command)
```

### Parameters

**command** ConditionalCommand

The conditional command to push.

### Exceptions

[ArgumentNullException](#)

Thrown when the command is null.

## Run()

Executes the stored program by iterating through commands and executing them.

```
public override void Run()
```

### Exceptions

StoredProgramException

Thrown when execution exceeds limits or errors occur during execution.

# Namespace ASE\_Assignment\_Test.Builder DesignPatternTest

## Classes

### [AppComponentBuilderTest](#)

Unit test class for AppComponentsBuilder.

### [AppComponentDirectorTest](#)

Unit test class for AppComponentsDirector.

# Class AppComponentBuilderTest

Namespace: [ASE Assignment Test.BuilderDesignPatternTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class for AppComponentsBuilder.

```
[TestClass]
public class AppComponentBuilderTest
```

Inheritance

[object](#) ← AppComponentBuilderTest

Inherited Members

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

## Methods

### Setup()

Initializes a new instance of the AppComponentsBuilder before each test.

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

### TestGetCanvas()

Test method to verify that BuildCanvas method returns an instance of AppCanvas.

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

### TestGetCommandFactory()

Test method to verify that BuildCommandFactory method returns an instance of AppCommandFactory.

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

## TestGetParser()

Test method to verify that BuildParser method returns an instance of AppParser.

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

## TestGetStoredProgram()

Test method to verify that BuildStoredProgram method returns an instance of AppStoredProgram.

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

# Class AppComponentDirectorTest

Namespace: [ASE Assignment Test.BuilderDesignPatternTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class for AppComponentsDirector.

```
[TestClass]  
public class AppComponentDirectorTest
```

Inheritance

[object](#) ← AppComponentDirectorTest

Inherited Members

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

## Methods

### Build\_CreatesAllComponents()

Test method to verify that the Build method creates all necessary components. This test ensures that after calling Build, all components (Canvas, CommandFactory, StoredProgram, Parser) are correctly created.

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

### Setup()

Initializes the test setup.

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

# Namespace ASE\_Assignment\_Test.Commands Test

## Classes

### [AppArrayTest](#)

Unit test class to verify the behavior of the AppArray command.

### [AppCircleTest](#)

Unit test class to verify the behavior of the AppCircle command.

### [AppElseTest](#)

Unit test class to verify the behavior of the AppElse command.

### [AppEndTest](#)

Unit test class to verify the behavior of the AppEnd command.

### [AppIfTest](#)

Unit test class to verify the behavior of the AppIf command.

### [AppIntTest](#)

Unit test class to verify the behavior of the AppInt command.

### [AppMethodTest](#)

Unit test class to verify the behavior of the AppMethod command.

### [AppRealTest](#)

Unit test class to verify the behavior of the AppReal command.

### [AppRectangleTest](#)

Unit test class to verify the behavior of the AppRectangle command.

### [AppTriangleTest](#)

Unit test class to verify the behavior of the AppTriangle command.

### [AppWhileTest](#)

Unit test class to verify the behavior of the AppWhile command.

### [AppWriteTest](#)

Unit test class to verify the behavior of the AppWrite command.

# Class AppArrayTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppArray command.

```
[TestClass]
public class AppArrayTest
```

Inheritance

[object](#) ← AppArrayTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestArrayCommand()

Test method to verify the correct execution of an array-related program command. This test checks if variables and arrays are correctly processed.

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

# Class AppCircleTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppCircle command.

```
[TestClass]
public class AppCircleTest
```

Inheritance

[object](#) ← AppCircleTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestCircleCommand()

Verifies the behavior of the circle command in terms of canvas position and pen color.

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

# Class AppElseTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppElse command.

```
[TestClass]
public class AppElseTest
```

Inheritance

[object](#) ← AppElseTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestElseCommand()

Tests the behavior of the "else" command by verifying that the correct branch of the "if-else" structure is executed.

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

# Class AppEndTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppEnd command.

```
[TestClass]
public class AppEndTest
```

Inheritance

[object](#) ← AppEndTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestEndCommand()

Tests the behavior of the "end" command by verifying that loops terminate correctly when the "end while" is encountered.

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

# Class AppIfTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppIf command.

```
[TestClass]
public class AppIfTest
```

Inheritance

[object](#) ← AppIfTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestIfCommand()

Tests the behavior of the "if" command by verifying that conditional logic works correctly. The test checks if the correct branch is executed based on the condition.

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

# Class AppIntTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppInt command.

```
[TestClass]
public class AppIntTest
```

Inheritance

[object](#) ← AppIntTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestIntCommand()

Tests the behavior of the "int" command by verifying that integer variables are correctly initialized and their values are accurately retrieved.

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

# Class AppMethodTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppMethod command.

```
[TestClass]  
public class AppMethodTest
```

Inheritance

[object](#) ← AppMethodTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestMethodCommand()

Tests the behavior of the "method" command by verifying that a method is correctly defined, called with parameters, and its return value is properly assigned to a variable.

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

# Class AppRealTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppReal command.

```
[TestClass]  
public class AppRealTest
```

Inheritance

[object](#) ← AppRealTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestRealCommand()

Tests the behavior of the "real" command by verifying the correct assignment and retrieval of real number variables.

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

# Class AppRectangleTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppRectangle command.

```
[TestClass]
public class AppRectangleTest
```

## Inheritance

[object](#) ← AppRectangleTest

## Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestRectangleCommand()

Tests the behavior of the "rect" command in the application. Verifies that a rectangle is drawn with the specified parameters, checks the resulting pixel color, validates the final canvas position, and ensures the pen color updates correctly.

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

# Class AppTriangleTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppTriangle command.

```
[TestClass]
public class AppTriangleTest
```

Inheritance

[object](#) ← AppTriangleTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestTriangleCommand()

Tests the behavior of the "tri" command in the application. Verifies that a triangle can be drawn with the specified parameters and ensures no exceptions are thrown during execution.

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

# Class AppWhileTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppWhile command.

```
[TestClass]
public class AppWhileTest
```

## Inheritance

[object](#) ← AppWhileTest

## Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestWhileCommand()

Tests the behavior of the "while" command in the application. Validates that a while loop runs the specified number of iterations

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

# Class AppWriteTest

Namespace: [ASE Assignment Test.CommandsTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppWrite command.

```
[TestClass]  
public class AppWriteTest
```

Inheritance

[object](#) ← AppWriteTest

Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestWriteCommand()

Verifies the behavior of the write command.

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

# Namespace ASE\_Assignment\_Test.CoreTest

## Classes

### [AppCanvasClassTest](#)

Unit tests for the [AppCanvas](#) class.

### [AppCanvasCommandTest](#)

Unit tests for the [AppCanvas](#) class using AppParser and AppStoredProgram.

### [AppCommandFactoryTest](#)

Unit test class to verify the behavior of the AppCommandFactory. This class ensures that the AppCommandFactory correctly creates command objects based on a provided command string.

### [AppParserTest](#)

Unit test class to verify the behavior of the AppParser. This class ensures that the AppParser correctly parses commands and programs.

### [AppStoredProgramTest](#)

Unit test class to verify the behavior of the AppStoredProgram. This class ensures that the AppStoredProgram correctly stores, pushes, pops commands, and executes the program.

# Class AppCanvasClassTest

Namespace: [ASE Assignment Test.CoreTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit tests for the [AppCanvas](#) class.

```
[TestClass]
public class AppCanvasClassTest
```

Inheritance

[object](#) ← AppCanvasClassTest

Inherited Members

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

## Methods

### Circle\_InvalidNegativeRadius\_ThrowsCanvasException()

Ensures that the [Circle\(int, bool\)](#) method throws an exception for an invalid negative radius.

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

### Circle\_ValidRadiusFilled\_DrawsFilledCircle()

Validates that the [Circle\(int, bool\)](#) method correctly draws a filled circle with valid radius.

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

### DrawCircle\_MoveTo\_DrawTo\_Validates()

Tests multiline operations: MoveTo, SetColour, Circle, and DrawTo.

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

## DrawRect\_WriteText\_Validates()

Tests multiline operations: MoveTo, SetColour, Rect, and WriteText.

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

## DrawTo\_InvalidXPosition\_ThrowsCanvasException()

Checks that the [DrawTo\(int, int\)](#) method throws an exception for an invalid X-coordinate.

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

## DrawTo\_InvalidYPosition\_ThrowsCanvasException()

Checks that the [DrawTo\(int, int\)](#) method throws an exception for an invalid Y-coordinate.

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

## DrawTo\_ValidPosition\_DrawsLineAndUpdatesCoordinates()

Validates that the [DrawTo\(int, int\)](#) method correctly draws a line and updates coordinates.

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

## MoveTo\_InvalidXPosition\_ThrowsCanvasException()

Checks that the [MoveTo\(int, int\)](#) method throws an exception for an invalid X-coordinate.

```
[TestMethod]
public void MoveTo_InvalidXPosition.ThrowsCanvasException()
```

## MoveTo\_InvalidYPosition.ThrowsCanvasException()

Checks that the [MoveTo\(int, int\)](#) method throws an exception for an invalid Ycoordinate.

```
[TestMethod]
public void MoveTo_InvalidYPosition.ThrowsCanvasException()
```

## MoveTo\_ValidPosition\_UpdatesCoordinates()

Validates that the [MoveTo\(int, int\)](#) method correctly updates the coordinates.

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

## Rect\_ValidDimensionsFilled\_DrawsFilledRectangle()

Checks that the [Rect\(int, int, bool\)](#) method correctly draws a filled rectangle.

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

## Rect\_ValidDimensionsNotFilled\_DrawsOutlinedRectangle()

Validates that the [Rect\(int, int, bool\)](#) method correctly draws an outlined rectangle with valid dimensions.

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

## SetColour\_ValidRGBValues\_SetsPenColour()

Checks that the [SetColour\(int, int, int\)](#) method sets the pen colour correctly.

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

## Setup()

Sets up the test environment by initializing the canvas and graphics objects.

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

## Tri\_InvalidDimensions\_ThrowsCanvasException()

Ensures that the [Tri\(int, int\)](#) method draws a triangle with valid dimensions.

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

## WriteText\_EmptyOrNullText\_ThrowsCanvasException()

Ensures that the [WriteText\(string\)](#) method throws a CanvasException when the input text is null or empty.

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

# Class AppCanvasCommandTest

Namespace: [ASE Assignment Test.CoreTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit tests for the [AppCanvas](#) class using AppParser and AppStoredProgram.

```
[TestClass]
public class AppCanvasCommandTest
```

## Inheritance

[object](#) ← AppCanvasCommandTest

## Inherited Members

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

## Methods

### DrawCircle\_MoveTo\_DrawTo\_Test()

Tests multiline operations: MoveTo, Pen, Circle, and DrawTo.

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

### DrawRect\_WriteText\_Test()

Tests multiline operations: MoveTo, Pen, Rect, and WriteText.

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

### DrawTo\_ValidPosition\_Test()

Validates that the [DrawTo\(int, int\)](#) method correctly draws a line and updates coordinates.

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

## MoveTo\_ValidPosition\_Test()

Validates that the [MoveTo\(int, int\)](#) method correctly updates the coordinates.

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

## Setup()

Sets up the test environment by initializing the canvas and graphics objects.

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

# Class AppCommandFactoryTest

Namespace: [ASE Assignment Test.CoreTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppCommandFactory. This class ensures that the AppCommandFactory correctly creates command objects based on a provided command string.

```
[TestClass]
public class AppCommandFactoryTest
```

## Inheritance

[object](#) ← AppCommandFactoryTest

## Inherited Members

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

## Methods

### Setup()

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

### TestMakeCommand(string, Type)

Tests the behavior of the AppCommandFactory by verifying that the correct command type is created based on the given command string.

```
[TestMethod]
[DataRow(new object?[] { "tri", typeof(AppTriangle) })]
[DataRow(new object?[] { "write", typeof(AppWrite) })]
[DataRow(new object?[] { "rect", typeof(AppRectangle) })]
[DataRow(new object?[] { "circle", typeof(AppCircle) })]
[DataRow(new object?[] { "int", typeof(AppInt) })]
[DataRow(new object?[] { "real", typeof(AppReal) })]
```

```
[DataRow(new object?[] { "array", typeof(AppArray) })]  
[DataRow(new object?[] { "while", typeof(AppWhile) })]  
[DataRow(new object?[] { "if", typeof(AppIf) })]  
[DataRow(new object?[] { "for", typeof(AppFor) })]  
[DataRow(new object?[] { "else", typeof(AppElse) })]  
[DataRow(new object?[] { "method", typeof(AppMethod) })]  
[DataRow(new object?[] { "end", typeof(AppEnd) })]  
public void TestMakeCommand(string commandType, Type expectedType)
```

## Parameters

commandType [string](#)

expectedType [Type](#)

## TestMakeCommand\_InvalidCommandType()

Tests the MakeCommand method by providing an invalid command string and verifying that a FactoryException is thrown.

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

# Class AppParserTest

Namespace: [ASE Assignment Test.CoreTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppParser. This class ensures that the AppParser correctly parses commands and programs.

```
[TestClass]  
public class AppParserTest
```

## Inheritance

[object](#) ← AppParserTest

## Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestParseCommand()

Tests the ParseCommand method by providing a valid command string and verifying that the returned command object is of the correct type.

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

## TestParseProgram\_InvalidSyntax()

Tests the ParseProgram method by providing an invalid program string and verifying that a BOOSEException is thrown.

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

# Class AppStoredProgramTest

Namespace: [ASE Assignment Test.CoreTest](#)

Assembly: ASE\_Assignment\_Test.dll

Unit test class to verify the behavior of the AppStoredProgram. This class ensures that the AppStoredProgram correctly stores, pushes, pops commands, and executes the program.

```
[TestClass]  
public class AppStoredProgramTest
```

## Inheritance

[object](#) ← AppStoredProgramTest

## Inherited Members

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

## Methods

### Setup()

Sets up the test environment, initializing the necessary objects for testing.

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

### TestStoredCommand\_PushandPop()

Tests the push and pop operations of the stored program by pushing a command onto the stack and verifying that it can be popped correctly.

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

## TestStoredCommand\_Run()

Tests the Run method of the stored program by parsing a program and executing it, ensuring that no exceptions are thrown and the program runs successfully.

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

# Namespace BOOSEWebApp.Controllers

## Classes

### [DrawingController](#)

Controller responsible for handling drawing operations and interactions with the application canvas.

# Class DrawingController

Namespace: [BOOSEWebApp.Controllers](#)

Assembly: BOOSEWebApp.dll

Controller responsible for handling drawing operations and interactions with the application canvas.

```
public class DrawingController : Controller, IActionFilter, IAsyncActionFilter,  
IFilterMetadata, IDisposable
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← DrawingController

## Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

## Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),  
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),  
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),  
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),  
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),  
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),  
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),  
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),  
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),  
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#), [Controller.TempData](#),  
[Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#), [ControllerBase.StatusCode\(int, object\)](#),  
[ControllerBase.Content\(string\)](#), [ControllerBase.Content\(string, string\)](#),  
[ControllerBase.Content\(string, string, Encoding\)](#),  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),  
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),  
[ControllerBase.RedirectPermanent\(string\)](#), [ControllerBase.RedirectPreserveMethod\(string\)](#),  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.LocalRedirect\(string\)](#),  
[ControllerBase.LocalRedirectPermanent\(string\)](#), [ControllerBase.LocalRedirectPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.RedirectToAction\(\)](#),  
[ControllerBase.RedirectToAction\(string\)](#), [ControllerBase.RedirectToAction\(string, object\)](#),  
[ControllerBase.RedirectToAction\(string, string\)](#),  
[ControllerBase.RedirectToAction\(string, string, object\)](#),  
[ControllerBase.RedirectToAction\(string, string, string\)](#),

[ControllerBase.RedirectToAction\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,  [ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?\)](#)  ,  [ControllerBase.EntityTagHeaderValue](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,

[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,  
 [ControllerBase.File\(string, string\)](#) ,  [ControllerBase.File\(string, string, bool\)](#) ,  
 [ControllerBase.File\(string, string, string\)](#) ,  [ControllerBase.File\(string, string, string, bool\)](#) ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string\)](#) ,  [ControllerBase.PhysicalFile\(string, string, bool\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,  
 [ControllerBase.Unauthorized\(\)](#) ,  [ControllerBase.Unauthorized\(object\)](#) ,  [ControllerBase.NotFound\(\)](#) ,  
 [ControllerBase.NotFound\(object\)](#) ,  [ControllerBase.BadRequest\(\)](#) ,  
 [ControllerBase.BadRequest\(object\)](#) ,  [ControllerBase.BadRequest\(ModelStateDictionary\)](#) ,  
 [ControllerBase.UnprocessableEntity\(\)](#) ,  [ControllerBase.UnprocessableEntity\(object\)](#) ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#) ,  [ControllerBase.Conflict\(\)](#) ,  
 [ControllerBase.Conflict\(object\)](#) ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#) ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#) ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) ,  [ControllerBase.ValidationProblem\(\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
 [ControllerBase.Created\(\)](#) ,  [ControllerBase.Created\(string, object\)](#) ,  
 [ControllerBase.Created\(Uri, object\)](#) ,  [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#) ,  [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) ,  [ControllerBase.Accepted\(\)](#) ,  
 [ControllerBase.Accepted\(object\)](#) ,  [ControllerBase.Accepted\(Uri\)](#) ,  [ControllerBase.Accepted\(string\)](#) ,  
 [ControllerBase.Accepted\(string, object\)](#) ,  [ControllerBase.Accepted\(Uri, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string\)](#) ,  [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#) ,  
 [ControllerBase.AcceptedAtRoute\(object\)](#) ,  [ControllerBase.AcceptedAtRoute\(string\)](#) ,  
 [ControllerBase.AcceptedAtRoute\(string, object\)](#) ,  [ControllerBase.AcceptedAtRoute\(object, object\)](#) ,

[ControllerBase.AcceptedAtRoute\(string, object, object\)](#) ,  [ControllerBase.Challenge\(\)](#) ,  
 [ControllerBase.Challenge\(params string\[\]\)](#) ,  [ControllerBase.Challenge\(AuthenticationProperties\)](#) ,  
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#) ,  [ControllerBase.Forbid\(\)](#) ,  
 [ControllerBase.Forbid\(params string\[\]\)](#) ,  [ControllerBase.Forbid\(AuthenticationProperties\)](#) ,  
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#) ,  
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#) ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#) ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#) ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#) ,  [ControllerBase.SignOut\(\)](#) ,  
 [ControllerBase.SignOut\(AuthenticationProperties\)](#) ,  [ControllerBase.SignOut\(params string\[\]\)](#) ,  
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryValidateModel\(object\)](#) ,  [ControllerBase.TryValidateModel\(object, string\)](#) ,  
 [ControllerBase.HttpContext](#) ,  [ControllerBase.Request](#) ,  [ControllerBase.Response](#) ,  
 [ControllerBase.RouteData](#) ,  [ControllerBase.ModelState](#) ,  [ControllerBase.ControllerContext](#) ,  
 [ControllerBase.MetadataProvider](#) ,  [ControllerBase.ModelBinderFactory](#) ,  [ControllerBase.Url](#) ,  
 [ControllerBase.ObjectValidator](#) ,  [ControllerBase.ProblemDetailsFactory](#) ,  [ControllerBase.User](#) ,  
 [ControllerBase.Empty](#) ,  [object.Equals\(object\)](#) ,  [object.Equals\(object, object\)](#) ,  
 [object.GetHashCode\(\)](#) ,  [object.GetType\(\)](#) ,  [object.MemberwiseClone\(\)](#) ,  
 [object.ReferenceEquals\(object, object\)](#) ,  [object.ToString\(\)](#)

## Constructors

### DrawingController()

Initializes a new instance of the [DrawingController](#) class. Sets up the core components using the Builder design pattern.

```
public DrawingController()
```

## Methods

### ExecuteCommand(string)

Executes a drawing command provided by the user.

```
[HttpPost]  
public IActionResult ExecuteCommand(string command)
```

#### Parameters

**command** [string](#)

The drawing command to be executed.

#### Returns

[IActionResult](#)

The updated Index view with execution results.

### Index()

Displays the main drawing interface.

```
[HttpGet]  
public IActionResult Index()
```

#### Returns

[IActionResult](#)

The Index view.

## Load(IFormFile)

Loads a file containing drawing commands and displays its content.

```
[HttpPost]  
public IActionResult Load(IFormFile uploadedFile)
```

### Parameters

[uploadedFile](#) [IFormFile](#)

The file containing drawing commands.

### Returns

[IActionResult](#)

The updated Index view with loaded file content.

## Reset()

Clears the canvas, resetting it to its initial state.

```
[HttpPost]  
public IActionResult Reset()
```

### Returns

[IActionResult](#)

The updated Index view with a cleared canvas.

# Namespace BOOSEWebApp.Models

## Classes

[ErrorViewModel](#)

# Class ErrorViewModel

Namespace: [BOOSEWebApp.Models](#)

Assembly: BOOSEWebApp.dll

```
public class ErrorViewModel
```

## Inheritance

[object](#) ← ErrorViewModel

## Inherited Members

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

# Properties

## RequestId

```
public string? RequestId { get; set; }
```

### Property Value

[string](#)

## ShowRequestId

```
public bool ShowRequestId { get; }
```

### Property Value

[bool](#)