

# Namespace ASE\_Assignment\_Bijesh

## Classes

### [MainForm](#)

The windows form which provides a GUI to use the boose interpreter and its implementations

# Class MainForm

Namespace: ASE Assignment Bijesh

Assembly: ASE Assignment Bijesh.dll

The windows form which provides a GUI to use the boose interpreter and its implementations

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

## Inheritance

```
object ↳ ← MarshalByRefObject ↳ ← Component ↳ ← Control ↳ ← ScrollableControl ↳ ←  
ContainerControl ↳ ← Form ↳ ← MainForm
```

## 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.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,  
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,  
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,  
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.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\(EventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,  
[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\(Control, bool, bool, bool, bool\)](#) , [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.IsAccessible](#) ,  
[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.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) ,  
[Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) ,  
[Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) ,  
[Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) ,  
[Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) ,  
[Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) ,  
[Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) ,  
[Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) ,  
[Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,  
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,  
[Component.Dispose\(\)](#) , [Component.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

### MainForm()

The public constructor for the form. Initializes all of the factory, program and parser objects

```
public MainForm()
```

## 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\_Bijesh.Src

## Classes

### [BooseSingleRunnerForTests](#)

An encapsulated streamlined class for testing the BOOSE interpreter

### [BooseWrapper](#)

An encapsulated streamlined class for running the BOOSE interpreter

## Interfaces

### [IBooseWrapper](#)

An interface to create a streamlined boose wrapper

# Class BooseSingleRunnerForTests

Namespace: [ASE Assignment Bijesh.Src](#)

Assembly: ASE Assignment Bijesh.dll

An encapsulated streamlined class for testing the BOOSE interpreter

```
public class BooseSingleRunnerForTests
```

## Inheritance

[object](#) ← BooseSingleRunnerForTests

## Inherited Members

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

## Constructors

### BooseSingleRunnerForTests()

Public constructor

```
public BooseSingleRunnerForTests()
```

## Methods

### GetCurrentPosition()

```
public Point GetCursorPosition()
```

Returns

[Point](#)

## RunProgram(string)

```
public void RunProgram(string program)
```

### Parameters

program [string](#)

# Class BooseWrapper

Namespace: [ASE Assignment Bijesh.Src](#)

Assembly: ASE Assignment Bijesh.dll

An encapsulated streamlined class for running the BOOSE interpreter

```
public class BooseWrapper : IBooseWrapper
```

Inheritance

[object](#) ← BooseWrapper

Implements

[IBooseWrapper](#)

Inherited Members

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

## Properties

Instance

```
public static BooseWrapper Instance { get; }
```

Property Value

[BooseWrapper](#)

## Methods

ClearScreen()

Clears the Screen

```
public void ClearScreen()
```

## GetBitmap()

Returns the bitmap of the current state of the canvas

```
public Bitmap GetBitmap()
```

Returns

[Bitmap](#)

## GetCurrentPosition()

Returns the current X,Y coordinate of the virtual cursor at

```
public Point GetCursorPosition()
```

Returns

[Point](#)

## GetInstance()

Returns the singleton instance as a IBooseWrapper reference

```
public static IBooseWrapper GetInstance()
```

Returns

[IBooseWrapper](#)

## Reset()

Clears the Screen

```
public void Reset()
```

## RunProgram(string)

Compiles and runs the program given in the program string

```
public void RunProgram(string program)
```

### Parameters

program string

# Interface IBooseWrapper

Namespace: [ASE Assignment Bijesh.Src](#)

Assembly: ASE Assignment Bijesh.dll

An interface to create a streamlined boose wrapper

```
public interface IBooseWrapper
```

## Fields

### Instance

Single static singleton instance

```
public static BooseWrapper Instance
```

Field Value

[BooseWrapper](#)

## Methods

### ClearScreen()

Clears the Screen

```
void ClearScreen()
```

### GetBitmap()

Returns the bitmap of the current state of the canvas

```
Bitmap GetBitmap()
```

Returns

[Bitmap](#)

## GetCurrentPosition()

Returns the current X,Y coordinate of the virtual cursor at

```
Point GetCurrentPosition()
```

Returns

[Point](#)

## GetInstance()

Returns the static instance

```
public static abstract IBooseWrapper GetInstance()
```

Returns

[IBooseWrapper](#)

## Reset()

Resets the interpreter to the starting condition

```
void Reset()
```

## RunProgram(string)

Compiles and runs the program given in the program string

```
void RunProgram(string program)
```

## Parameters

program string ↗

# Namespace ASE\_Assignment\_Bijesh.Src. Command

## Classes

### [CommandBrushColor](#)

The brush color object returned for the brush command, changes the brush color on execution

### [CommandCircle](#)

The circle object returned for the circle command. Draws a circle on execution. May be a filled circle if true is passed as second parameter.

### [CommandFilledCircle](#)

The filled circle object returned for the fcircle command. Draws a filled circle on execution.

### [CommandFilledRectangle](#)

The filled rectangle object returned for the fcircle command. Draws a filled rectangle on execution.

### [CommandRect](#)

The rectangle object returned for the circle command. Draws a rectangle on execution. May be a filled rectangle if true is passed as second parameter.

### [CommandTriangle](#)

The triangle object returned for the tri command. Draws an Isosceles triangle on execution.

### [CommandWrite](#)

The write object returned for the write command.

# Class CommandBrushColor

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The brush color object returned for the brush command, changes the brush color on execution

```
public class CommandBrushColor : CommandThreeParameters, ICommand
```

## Inheritance

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

## 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

### CommandBrushColor()

Public empty constructor

```
public CommandBrushColor()
```

## CommandBrushColor(Canvas, int, int, int)

Parameterized constructor to directly create object without factory interaction

```
public CommandBrushColor(Canvas canvas, int r, int g, int b)
```

### Parameters

**canvas** Canvas

The canvas object of the output.

**r** [int](#)

The red value.

**g** [int](#)

The green value.

**b** [int](#)

The blue value.

## Methods

### CheckParameters(string[])

Validation of the parameters

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

### Parameters

**parameterList** [string](#)[]

The string array of parameters.

### Execute()

The Execute method called during stored program execution

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

# Class CommandCircle

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The circle object returned for the circle command. Draws a circle on execution. May be a filled circle if true is passed as second parameter.

```
public class CommandCircle : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

## CommandCircle()

```
public CommandCircle()
```

# Methods

## CheckParameters(string[])

Validation of the parameters

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

Parameters

parameterList [string](#)[]

The string array of parameters.

## Execute()

The Execute method called during stored program execution

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

# Class CommandFilledCircle

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The filled circle object returned for the fcircle command. Draws a filled circle on execution.

```
public class CommandFilledCircle : CommandOneParameter, ICommand
```

## Inheritance

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

## 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

### CommandFilledCircle()

Public empty constructor

```
public CommandFilledCircle()
```

### CommandFilledCircle(Canvas, int)

Parameterized constructor to directly create object without factory interaction

```
public CommandFilledCircle(Canvas canvas, int radius)
```

## Parameters

**canvas** Canvas

The canvas object of the output.

**radius** [int](#)

The radius of the circle.

## Methods

**CheckParameters(string[])**

Validation of the parameters

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

## Parameters

**parameterList** [string](#)[]

The string array of parameters.

**Execute()**

The Execute method called during stored program execution

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

# Class CommandFilledRectangle

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The filled circle object returned for the fcircle command. Draws a filled circle on execution.

```
public class CommandFilledRectangle : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

### CommandFilledRectangle()

Public empty constructor

```
public CommandFilledRectangle()
```

### CommandFilledRectangle(Canvas, int, int)

Parameterized constructor to directly create object without factory interaction

```
public CommandFilledRectangle(Canvas canvas, int width, int height)
```

## Parameters

**canvas** Canvas

The canvas object of the output.

**width** [int](#)

The width of the rectangle.

**height** [int](#)

The height of the rectangle.

## Methods

### CheckParameters(string[])

Validation of the parameters

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

## Parameters

**parameterList** [string](#)[]

The string array of parameters.

### Execute()

The Execute method called during stored program execution

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

# Class CommandRect

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The circle object returned for the circle command. Draws a circle on execution. May be a filled circle if true is passed as second parameter.

```
public class CommandRect : CommandThreeParameters, ICommand
```

## Inheritance

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

## 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

### CommandRect()

```
public CommandRect()
```

## Methods

## CheckParameters(string[])

Validation of the parameters

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

Parameters

parameterList [string](#)[]

The string array of parameters.

## Execute()

The Execute method called during stored program execution

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

# Class CommandTriangle

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The triangle object returned for the tri command. Draws an Isosceles triangle on execution.

```
public class CommandTriangle : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

### CommandTriangle()

Public empty constructor

```
public CommandTriangle()
```

### CommandTriangle(Canvas, int, int)

Parameterized constructor to directly create object without factory interaction

```
public CommandTriangle(Canvas canvas, int width, int height)
```

## Parameters

**canvas** Canvas

The canvas object of the output.

**width** int

The width of the base of the triangle.

**height** int

The height of the triangle.

## Methods

### CheckParameters(string[])

Validation of the parameters

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

## Parameters

**parameterList** string[]

The string array of parameters.

### Execute()

The Execute method called during stored program execution

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

# Class CommandWrite

Namespace: [ASE Assignment Bijesh.Src.Command](#)

Assembly: ASE Assignment Bijesh.dll

The write object returned for the write command.

```
public class CommandWrite : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

## CommandWrite()

Public empty constructor

```
public CommandWrite()
```

## CommandWrite(Canvas)

Parameterized constructor to directly create object without factory interaction

```
public CommandWrite(Canvas canvas)
```

Parameters

canvas Canvas

## Methods

### CheckParameters(string[])

Validation of the parameters

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

Parameters

parameterList string[]

### Execute()

The Execute method called during stored program execution Writes the given expression after it is evaluated by the stored program

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

# Namespace ASE\_Assignment\_Bijesh.Src. Components

## Classes

### [AppCanvas](#)

The canvas class which is to be manipulated by the BOOSE interpreter

### [AppCommand](#)

Extension of the app command class, Unused, only created for study

### [AppCommandFactory](#)

Uses the factory design methodology. The command factory which takes in the parsed command from the parser and then returns new objects specified by the commands

### [AppParser](#)

Replaced parser for this implementation of the BOOSE interpreter.

### [AppStoredProgram](#)

Replaced Implementation of the stored program class

## Interfaces

### [IExtendedCanvas](#)

Extended canvas interface to support additional features.

# Class AppCanvas

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

The canvas class which is to be manipulated by the BOOSE interpreter

```
public class AppCanvas : IExtendedCanvas, ICanvas
```

Inheritance

[object](#) ← AppCanvas

Implements

[IExtendedCanvas](#), ICanvas

Inherited Members

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

## Constructors

### AppCanvas()

Public constructor that calls the set function to setup the canvas

```
public AppCanvas()
```

## Properties

### BrushColour

Return and set the brush colour

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

### Property Value

[object](#)

## PenColour

Required Pen object from the interface, unused.

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

### Property Value

[object](#)

## Xpos

X coordinate of the current position

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

### Property Value

[int](#)

## Ypos

Y coordinate of the current position

```
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](#)

The Radius of the circle.

**filled** [bool](#)

The fill status of the circle if true, filled.

## Clear()

Clears the screen by filling it with gray

```
public void Clear()
```

## DrawTo(int, int)

Draws a line from the current position to the given end point

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

Parameters

**x** [int](#)

The x coordinate of the end point.

**y** [int](#)

The y coordinate of the end point.

## MoveTo(int, int)

Sets the current position to the given point

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

Parameters

x [int](#)

The x coordinate of the given point.

y [int](#)

The y coordinate of the given point.

## Rect(int, int, bool)

Draws a rectangle at the current position of the given parameters.

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

Parameters

width [int](#)

The width of the rectangle.

height [int](#)

The height of the rectangle.

filled [bool](#)

Boolean value to set the fill of the rectangle.

## Reset()

Resets the current position of the cursor.

```
public void Reset()
```

## Set(int, int)

Initializer for the app canvas class. Sets the canvas size, brush and color.

```
public void Set(int xsize, int ysize)
```

### Parameters

xsize [int ↗](#)

The width of the canvas.

ysize [int ↗](#)

The height of the canvas.

## SetBrushColour(int, int, int)

Changes the color of the brush given an rgb value.

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

### Parameters

red [int ↗](#)

The red value.

green [int ↗](#)

The green value.

blue [int ↗](#)

The blue value.

## SetColour(int, int, int)

Changes the color of the pen given an rgb value.

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

## Parameters

red [int](#)

The red value.

green [int](#)

The green value.

blue [int](#)

The blue value.

## Tri(int, int)

Draws a triangle at the current position with the given width and height

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

## Parameters

width [int](#)

height [int](#)

## WriteText(string)

Writes a string on the canvas at the current position

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

## Parameters

text [string](#)

The string value to be printed.

## getBitmap()

Returns the bitmap of the canvas

```
public object getBitmap()
```

Returns

object

Bitmap

# Class AppCommand

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

Extension of the app command class, Unused, only created for study

```
public abstract class AppCommand : Command, ICommand
```

Inheritance

[object](#) ← Command ← AppCommand

Implements

ICommand

Inherited Members

Command.program , Command.parameterList , Command.parameters , Command.paramsint ,  
[Command.CheckParameters\(string\[\]\)](#) , [Command.Set\(StoredProgram, string\)](#) , Command.Compile() ,  
Command.Execute() , [Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program ,  
Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

## Constructors

### AppCommand(AppStoredProgram)

Primary constructor

```
protected AppCommand(AppStoredProgram storedProgram)
```

Parameters

storedProgram [AppStoredProgram](#)

## Fields

## \_storedProgram

The stored program reference used to reference to the command stack

```
protected AppStoredProgram _storedProgram
```

### Field Value

[AppStoredProgram](#)

## Properties

### StoredProgram

Public setter and getter to access the Programs store

```
public AppStoredProgram StoredProgram { get; set; }
```

### Property Value

[AppStoredProgram](#)

## Methods

### Set(AppStoredProgram, string)

Function to add the current command to the stored program

```
public void Set(AppStoredProgram Program, string Params)
```

### Parameters

Program [AppStoredProgram](#)

Params [string](#)

# Class AppCommandFactory

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

Uses the factory design methodology. The command factory which takes in the parsed command from the parser and then returns new objects specified by the commands

```
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)

Method to return the new required objects

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

Parameters

commandType [string](#)

Returns

ICommand

# Class AppParser

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

Replaced parser for this implementation of the BOOSE interpreter.

```
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(CommandFactory, AppStoredProgram)

The public constructor to initialize the parser with its factory and stored program

```
public AppParser(CommandFactory Factory, AppStoredProgram Program)
```

## Parameters

**Factory** CommandFactory

**Program** [AppStoredProgram](#)

## Methods

### ParseCommand(string)

Parses the passed in single line BOOSE code into a line of BOOSE program and adds it to the list of stored programs

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

Parameters

line string

One line of preprocessed BOOSE program

Returns

ICommand

Exceptions

ParserException

## ParseProgram(string)

Takes in a multi-line BOOSE program string and converts it single line programs, converting any evaluations into their respective base commands and passes it to the Parse command method

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

Parameters

program string

# Class AppStoredProgram

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

Replaced Implementation of the stored program class

```
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.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.Commandsleft() ,  
StoredProgram.PC , [ArrayList.Adapter\(IList\)](#) , [ArrayList.Add\(object\)](#) ,  
[ArrayList.AddRange\(ICollection\)](#) , [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\(int, ICollection\)](#) , [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\(int, ICollection\)](#) ,  
[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.IsFixedSize](#) , [ArrayList.IsReadOnly](#) , [ArrayList.IsSynchronized](#) , [ArrayList.this\[int\]](#) ,  
[ArrayList.SyncRoot](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Remarks

Default constructor, just initializes the base class

## Constructors

### AppStoredProgram(ICanvas)

Replaced Implementation of the stored program class

```
public AppStoredProgram(ICanvas canvas)
```

Parameters

`canvas` ICanvas

## Remarks

Default constructor, just initializes the base class

## Methods

### AddMethod(EvaluationMethod)

Add a method into the methods list TODO: overwrite or replace methods of the same name for the latter declaration

```
public void AddMethod(EvaluationMethod method)
```

Parameters

method [EvaluationMethod](#)

## GetMethod(string)

Finds a method by its name

```
public EvaluationMethod GetMethod(string MethodName)
```

Parameters

MethodName [string](#)

Returns

[EvaluationMethod](#)

The method

Exceptions

StoredProgramException

## Pop()

Remove the top program from the program conditional commands stack

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

Returns

ConditionalCommand

BOOSE Conditional Command Object

Exceptions

StoredProgramException

## Push(ConditionalCommand)

Insert a Conditional command into the command program stack

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

### Parameters

Com ConditionalCommand

## ResetProgram()

Resets the program to the initial state

```
public override void ResetProgram()
```

## Run()

Run the program using the compiled programs list

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

## Exceptions

StoredProgramException

# Interface IExtendedCanvas

Namespace: [ASE Assignment Bijesh.Src.Components](#)

Assembly: ASE Assignment Bijesh.dll

Extended canvas interface to support additional features.

```
public interface IExtendedCanvas : ICanvas
```

## Inherited Members

[ICanvas.Set\(int, int\)](#) , [ICanvas.SetColour\(int, int, int\)](#) , [ICanvas.MoveTo\(int, int\)](#) ,  
[ICanvas.DrawLine\(int, int\)](#) , [ICanvas.DrawTo\(int, int\)](#) , [ICanvas.Clear\(\)](#) , [ICanvas.Reset\(\)](#) , [ICanvas.Circle\(int, bool\)](#) ,  
[ICanvas.Rect\(int, int, bool\)](#) , [ICanvas.Tri\(int, int\)](#) , [ICanvas.WriteText\(string\)](#) , [ICanvas.getBitmap\(\)](#) ,  
ICanvas.Xpos , ICanvas.Ypos , ICanvas.PenColour

## Properties

### BrushColour

Return and set the brush colour

```
object BrushColour { get; set; }
```

### Property Value

[object](#)

## Methods

### SetBrushColour(int, int, int)

Change the color of the brush used for text and filled shapes

```
void SetBrushColour(int red, int green, int blue)
```

### Parameters

red [int ↗](#)

green [int ↗](#)

blue [int ↗](#)

# Namespace ASE\_Assignment\_Bijesh.Src. Conditionals

## Classes

### [AppCompoundCommand](#)

The replaced Compound command implementation from the BOOSE library, Currently unused, was just used for study

### [AppConditionalCommand](#)

The replaced conditional command implementation from the BOOSE library, Currently unused, was just used for study

## Enums

### [AppConditionalCommand.ConditionalTypes](#)

Specifies the conditional types

# Class AppCompoundCommand

Namespace: [ASE Assignment Bijesh.Src.Conditionals](#)

Assembly: ASE Assignment Bijesh.dll

The replaced Compound command implementation from the BOOSE library, Currently unused, was just used for study

```
public class AppCompoundCommand : AppConditionalCommand, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← [AppConditionalCommand](#) ← AppCompoundCommand

## Implements

ICommand

## Inherited Members

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\(object, object\)](#)

## Methods

### Compile()

Compiles the conditional command and its it on the program stack

```
public override void Compile()
```

## Execute()

Execute method which runs during stored program execution

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

## Restrictions()

Remove the restrictions

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

# Class AppConditionalCommand

Namespace: [ASE Assignment Bijesh.Src.Conditionals](#)

Assembly: ASE Assignment Bijesh.dll

The replaced conditional command implementation from the BOOSE library, Currently unused, was just used for study

```
public class AppConditionalCommand : ConditionalCommand, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← AppConditionalCommand

## Implements

ICommand

## Derived

[AppCompoundCommand](#)

## Inherited Members

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\(object, object\)](#)

## Methods

### Compile()

Compiles the program and pushes it into the stack

```
public override void Compile()
```

## Execute()

The execute function which runs during program execution

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

## Restrictions()

Removes the restrictions of the base class

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

# Enum AppConditionalCommand.ConditionalTypes

Namespace: [ASE Assignment Bijesh.Src.Conditionals](#)

Assembly: ASE Assignment Bijesh.dll

Specifies the conditional types

```
public enum AppConditionalCommand.ConditionalTypes
```

## Fields

commFor = 2

commIF = 0

commWhile = 1

# Namespace ASE\_Assignment\_Bijesh.Src. Evaluations

## Classes

### [EvaluationArray](#)

The replaced implementation of the array class. Accepts an array of type Int and Real Only reduces/removes the restrictions for now.

### [EvaluationBool](#)

Replaced implementation of the Boolean class Just removes the number of instances limit restriction

### [EvaluationCall](#)

Replaced Implementation of the call function to support the rewritten Evaluation Methods class

### [EvaluationElse](#)

Replaced implementation of the else command, reduces the restrictions. Must be used alongside an if command

### [EvaluationEnd](#)

Replaced implementation of the end command, reduces the restrictions. Must be used at the end of a conditional or a method command block

### [EvaluationFor](#)

Replaced implementation of the for command, reduces the restrictions. Allows the program to create a for loop

### [EvaluationIf](#)

Replaced implementation of the if command, reduces the restrictions. Allows conditional branching

### [EvaluationInt](#)

Replaced implementation of the Int command, reduces the restrictions. Used to store and retrieve integer variables and their values.

### [EvaluationIntRewrite](#)

Rewrite of the int class without inheriting Int, not used due to issues with the stored program

### [EvaluationMethod](#)

### [EvaluationReal](#)

Replaced implementation of the Real command, reduces the restrictions. Used to store and retrieve real number variables and their values.

### [EvaluationWhile](#)

Replaced implementation of the While command, reduces the restrictions. Used to create while loops in the interpreter. Needs to be accompanied by an end command

# Class EvaluationArray

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

The replaced implementation of the array class. Accepts an array of type Int and Real Only reduces/removes the restrictions for now.

```
public class EvaluationArray : Array, ICommand
```

## Inheritance

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

## 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

### EvaluationArray()

Public Constructor, just reduces the restriction counter.

```
public EvaluationArray()
```

# Class EvaluationBool

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the Boolean class Just removes the number of instances limit restriction

```
public class EvaluationBool : Boolean, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← EvaluationBool

## Implements

ICommand

## Inherited Members

Boolean.Compile() , Boolean.Execute() , 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\(object, object\)](#)

# Methods

## Restrictions()

Removes the restrictions on the base boolean class by overriding it

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

# Class EvaluationCall

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced Implementation of the call function to support the rewritten Evaluation Methods class

```
public class EvaluationCall : Call, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Call ← EvaluationCall

## Implements

ICommand

## Inherited Members

Call.methodName , 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

## EvaluationCall()

The empty constructor, upcasts the base stored program object into an AppStoredProgram object.

```
public EvaluationCall()
```

# Methods

## Compile()

Creates method objects to store in the program stack. Reduces the restriction counter of the base class.

```
public override void Compile()
```

## Execute()

The Execute method called during stored program execution finds and executes the method as a program.

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

## Restrictions()

Just removes the restrictions from the compound command class

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

# Class EvaluationElse

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the else command, reduces the restrictions. Must be used alongside an if command

```
public class EvaluationElse : Else, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

[Else.CheckParameters\(string\[\]\)](#) , Else.Compile() , Else.Execute() , Else.CorrectingEnd ,  
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

### EvaluationElse()

Public Constructor, just reduces the restriction counter.

```
public EvaluationElse()
```

## Methods

### Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationEnd

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the end command, reduces the restrictions. Must be used at the end of a conditional or a method command block

```
public class EvaluationEnd : End, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

End.Compile() , CompoundCommand.ReduceRestrictions() ,  
[CompoundCommand.CheckParameters\(string\[\]\)](#) , 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

### EvaluationEnd()

Public Constructor, just reduces the restriction counter.

```
public EvaluationEnd()
```

## Methods

### Execute()

Method called when the End method is executed in the BOOSE Program

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

### Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationFor

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the for command, reduces the restrictions. Allows the program to create a for loop

```
public class EvaluationFor : For, ICommand
```

## Inheritance

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

## 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\(object, object\)](#)

# Methods

## Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationIf

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the if command, reduces the restrictions. Allows conditional branching

```
public class EvaluationIf : If, ICommand
```

## Inheritance

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

## 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

### EvaluationIf()

Public Constructor, just reduces the restriction counter.

```
public EvaluationIf()
```

## Methods

### Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationInt

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the Int command, reduces the restrictions. Used to store and retrieve integer variables and their values.

```
public class EvaluationInt : Int, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

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\(object, object\)](#)

# Methods

## Compile()

Compiles the Int into the program stack

```
public override void Compile()
```

## Execute()

The Execute method called during program execution, evaluates and attempts to save an int variable

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

## Exceptions

StoredProgramException

## Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationIntRewrite

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Rewrite of the int class without inheriting Int, not used due to issues with the stored program

```
public class EvaluationIntRewrite : Evaluation, ICommand
```

## Inheritance

[object](#) ← Command ← Evaluation ← EvaluationIntRewrite

## Implements

ICommand

## Inherited Members

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

### EvaluationIntRewrite()

Public constructor

```
public EvaluationIntRewrite()
```

## Methods

### Compile()

Adds the program to the program stack to compile it

```
public override void Compile()
```

## Execute()

The method called during execution of the program

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

## Exceptions

StoredProgramException

# Class EvaluationMethod

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

```
public class EvaluationMethod : CompoundCommand, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

CompoundCommand.ReduceRestrictions() , [CompoundCommand.CheckParameters\(string\[\]\)](#) ,  
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

### EvaluationMethod()

```
public EvaluationMethod()
```

## Properties

## LocalVariables

The array to store the local variables

```
public string[] LocalVariables { get; }
```

Property Value

[string](#)[]

## MethodName

Public access to the private method name

```
public string MethodName { get; }
```

Property Value

[string](#)[]

## Type

The return type of the program

```
public string Type { get; }
```

Property Value

[string](#)[]

## Methods

### Compile()

Compiles the method and sets the beginning and returning line numbers and sets up the local variables

```
public override void Compile()
```

## Exceptions

CommandException

## Execute()

The method that runs during program execution

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

## Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationReal

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the Real command, reduces the restrictions. Used to store and retrieve real number variables and their values.

```
public class EvaluationReal : Real, ICommand
```

## Inheritance

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

## 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\(object, object\)](#)

# Methods

## Restrictions()

Removes the restrictions on the base class by overriding it

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

# Class EvaluationWhile

Namespace: [ASE Assignment Bijesh.Src.Evaluations](#)

Assembly: ASE Assignment Bijesh.dll

Replaced implementation of the While command, reduces the restrictions. Used to create while loops in the interpreter. Needs to be accompanied by an end command

```
public class EvaluationWhile : While, ICommand
```

## Inheritance

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

## 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

### EvaluationWhile()

Public Constructor, just reduces the restriction counter.

```
public EvaluationWhile()
```

## Methods

### Restrictions()

Removes the restrictions on the base class by overriding it

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

# Namespace ASE\_Assignment\_Bijesh\_Tests

## Classes

### [CommandCircleTests](#)

The Unit tests for the CommandCircle Class

### [MainAppTests](#)

The Unit tests for the Mainform Class but tests the overall BOOSE System implementation.

# Class CommandCircleTests

Namespace: [ASE Assignment Bijesh Tests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandCircle Class

```
[TestClass]
public class CommandCircleTests
```

Inheritance

[object](#) ← CommandCircleTests

Inherited Members

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

## Methods

### CheckParameters\_InvalidParameterCount\_ThrowsException()

Checks if wrong parameter count throws exception

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

### CheckParameters\_ValidParameterCount\_DoesNotThrowException()

Checks if correct parameter doesn't throw exception

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

# Class MainAppTests

Namespace: [ASE Assignment Bijesh Tests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the Mainform Class but tests the overall BOOSE System implementation.

```
[TestClass]  
public class MainAppTests
```

Inheritance

[object](#) ← MainAppTests

Inherited Members

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

## Methods

### TestDrawToMainProgram()

Tests the drawto command passed through the parser

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

### TestMoveToMainProgram()

Tests the moveto command passed through the parser

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

### TestMultiLineMainProgram()

Tests system by passing in a multiline program

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

# Namespace ASE\_Assignment\_Tests.Command Tests

## Classes

### [CommandBrushColorTests](#)

The Unit tests for the CommandBrushColor Class

### [CommandFilledCircleTests](#)

The Unit tests for the CommandFilledCircle Class

### [CommandFilledRectangleTests](#)

The Unit tests for the CommandFilledRectangle Class

### [CommandRectTests](#)

The Unit tests for the CommandRect Class

### [CommandTriangleTests](#)

The Unit tests for the CommandTriangle Class

### [CommandWriteTests](#)

The Unit tests for the CommandWrite Class

### [EvaluationArrayTests](#)

The Unit tests for the EvaluationArray Class

### [EvaluationBoolTests](#)

The Unit tests for the EvaluationBool Class

### [EvaluationElseTests](#)

The Unit tests for the EvaluationElse Class

### [EvaluationEndTests](#)

The Unit tests for the EvaluationEnd Class

### [EvaluationForTests](#)

The Unit tests for the EvaluationFor Class

### [EvaluationIfTests](#)

The Unit tests for the EvaluationIf Class

### [EvaluationIntTests](#)

The Unit tests for the CommandCall Class

## EvaluationMethodTests

The Unit tests for the CommandCall Class

## EvaluationRealTests

The Unit tests for the EvaluationReal Class

## EvaluationWhileTests

The Unit tests for the CommandCall Class

# Class CommandBrushColorTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandBrushColor Class

```
[TestClass]
public class CommandBrushColorTests
```

Inheritance

[object](#) ← CommandBrushColorTests

Inherited Members

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

## Methods

### CheckParameters\_InvalidParameterCount\_ThrowsException()

Checks if wrong parameter count throws exception

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

### CheckParameters\_ValidParameterCount\_DoesNotThrowException()

Checks if correct parameter doesn't throw exception

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

# Class CommandFilledCircleTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandFilledCircle Class

```
[TestClass]
public class CommandFilledCircleTests
```

Inheritance

[object](#) ← CommandFilledCircleTests

Inherited Members

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

## Methods

[CheckParameters\\_InvalidParameterCount\\_ThrowsException\(\)](#)

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

[CheckParameters\\_ValidParameterCount\\_DoesNotThrowException\(\)](#)

Checks if correct parameter doesn't throw exception

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

# Class CommandFilledRectangleTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandFilledRectangle Class

```
[TestClass]
public class CommandFilledRectangleTests
```

Inheritance

[object](#) ← CommandFilledRectangleTests

Inherited Members

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

## Methods

**CheckParameters\_InvalidParameterCount\_ThrowsException()**

Checks if wrong parameter count throws exception

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

**CheckParameters\_ValidParameterCount\_DoesNotThrowException()**

Checks if correct parameter doesn't throw exception

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

# Class CommandRectTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandRect Class

```
[TestClass]
public class CommandRectTests
```

Inheritance

[object](#) ← CommandRectTests

Inherited Members

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

## Methods

### CheckParameters\_InvalidParameterCount\_ThrowsException()

Checks if wrong parameter count throws exception

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

### CheckParameters\_ValidParameterCount\_DoesNotThrowException()

Checks if correct parameter doesn't throw exception

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

# Class CommandTriangleTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandTriangle Class

```
[TestClass]
public class CommandTriangleTests
```

Inheritance

[object](#) ← CommandTriangleTests

Inherited Members

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

## Methods

**CheckParameters\_InvalidParameterCount\_ThrowsException()**

Checks if wrong parameter count throws exception

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

**CheckParameters\_ValidParameterCount\_DoesNotThrowException()**

Checks if correct parameter doesn't throw exception

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

# Class CommandWriteTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandWrite Class

```
[TestClass]  
public class CommandWriteTests
```

Inheritance

[object](#) ← CommandWriteTests

Inherited Members

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

## Methods

### CheckParameters\_InvalidParameterCount\_ThrowsException()

Checks if wrong parameter count throws exception

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

### CheckParameters\_ValidParameterCount\_DoesNotThrowException()

Checks if correct parameter doesn't throw exception

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

# Class EvaluationArrayTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationArray Class

```
[TestClass]
public class EvaluationArrayTests
```

Inheritance

[object](#) ← EvaluationArrayTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

# Class EvaluationBoolTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationBool Class

```
[TestClass]
public class EvaluationBoolTests
```

Inheritance

[object](#) ← EvaluationBoolTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the class objects can be initialized

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

# Class EvaluationElseTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationElse Class

```
[TestClass]
public class EvaluationElseTests
```

Inheritance

[object](#) ← EvaluationElseTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

# Class EvaluationEndTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationEnd Class

```
[TestClass]
public class EvaluationEndTests
```

Inheritance

[object](#) ← EvaluationEndTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### MatchedEndDoesNotThrowException()

Checks if matched end doesn't throw exception

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

## UnmatchedEndThrowsException()

Checks if unmatched ends throw exception

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

# Class EvaluationForTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationFor Class

```
[TestClass]  
public class EvaluationForTests
```

Inheritance

[object](#) ← EvaluationForTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### CorrectForDoesNotThrowException()

Checks correct for declaration doesn't throw exception

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

## IncorrectForThrowsException()

Checks if incorrect for declaration throw exception

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

# Class EvaluationIfTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationIf Class

```
[TestClass]
public class EvaluationIfTests
```

Inheritance

[object](#) ← EvaluationIfTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### MatchedIfDoesNotThrowException()

Checks if matched if doesn't throw exception

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

## UnmatchedIfThrowsException()

Checks if unmatched if throw exception

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

# Class EvaluationIntTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandCall Class

```
[TestClass]
public class EvaluationIntTests
```

Inheritance

[object](#) ← EvaluationIntTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### InvalidValueThrowsException()

Checks if invalid value throws exception

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

## ValidValueDoesNotThrowException()

Checks if valid value doesn't throw exception

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

# Class EvaluationMethodTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandCall Class

```
[TestClass]
public class EvaluationMethodTests
```

Inheritance

[object](#) ← EvaluationMethodTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### InvalidDeclarationThrowsException()

Checks if invalid declaration throws exception

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

## ValidDeclarationDoesNotThrowException()

Checks if valid declaration end doesn't throw exception

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

# Class EvaluationRealTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationReal Class

```
[TestClass]
public class EvaluationRealTests
```

Inheritance

[object](#) ← EvaluationRealTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### InvalidValueThrowsException()

Checks if invalid value throws exception

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

## ValidValueDoesNotThrowException()

Checks if valid value doesn't throw exception

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

# Class EvaluationWhileTests

Namespace: [ASE Assignment Tests.CommandTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandCall Class

```
[TestClass]
public class EvaluationWhileTests
```

Inheritance

[object](#) ← EvaluationWhileTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### InvalidDeclarationThrowsException()

Checks if incorrect declaration throw exception

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

## ValidDeclarationDoesNotThrowException()

Checks correct for declaration doesn't throw exception

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

# Namespace ASE\_Assignment\_Tests.Component Tests

## Classes

### [AppCanvasTests](#)

The Unit tests for the AppCanvas Class. Tests the final positions of the cursor after the methods are called.

### [AppCommandFactoryTests](#)

The Unit tests for the CommandFactory Class. Checks the factory return types.

### [AppParserTests](#)

The Unit tests for the AppParser Class. Tests the returned objects and program parsing.

### [AppStoredProgramTests](#)

The Unit tests for the AppStoredProgram Class. Checks the exceptions from the class.

# Class AppCanvasTests

Namespace: [ASE Assignment Tests.ComponentTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the AppCanvas Class. Tests the final positions of the cursor after the methods are called.

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

## Inheritance

[object](#) ← AppCanvasTests

## Inherited Members

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

# Methods

## TestBitmap()

Tests the get bitmap function

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

## TestChangeBrushColour()

Tests the changing the brush colour externally

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

## TestChangePenColour()

Tests the changing the pen colour externally

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

## TestCircle()

Tests the circle method

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

## TestCircleFilled()

Tests the circle method with fill true

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

## TestClear()

Tests the canvas clear method

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

## TestDrawToCanvas()

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

## TestMoveToCanvas()

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

## TestMultiLineMainProgram()

Tests system by passing in a multiline program

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

## TestRect()

Tests the Rectangle method

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

## TestRectFilled()

Tests the Rectangle method with fill true

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

## TestResetCanvas()

Tests the canvas reset method

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

## TestSetCanvas()

Tests the set canvas method

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

## TestTriangle()

Tests the Triangle

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

## TestWriteText()

Tests the write text method

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

# Class AppCommandFactoryTests

Namespace: [ASE Assignment Tests.ComponentTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the CommandFactory Class. Checks the factory return types.

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

## Inheritance

[object](#) ← AppCommandFactoryTests

## Inherited Members

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

## Methods

### CheckReturnTypeBrush()

Checks if the Command Factory properly returns Brush object

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

### CheckReturnTypeCircle()

Checks if the Command Factory properly returns Circle object

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

### CheckReturnTypeEvaluationArray()

Checks if the Command Factory properly returns EvaluationArray object

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

## CheckReturnTypeEvaluationCall()

Checks if the Command Factory properly returns EvaluationCall object

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

## CheckReturnTypeEvaluationElse()

Checks if the Command Factory properly returns EvaluationElse object

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

## CheckReturnTypeEvaluationEnd()

Checks if the Command Factory properly returns EvaluationEnd object

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

## CheckReturnTypeEvaluationFor()

Checks if the Command Factory properly returns EvaluationFor object

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

## CheckReturnTypeEvaluationIf()

Checks if the Command Factory properly returns EvaluationIf object

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

## CheckReturnTypeEvaluationInt()

Checks if the Command Factory properly returns EvaluationInt object

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

## CheckReturnTypeEvaluationMethod()

Checks if the Command Factory properly returns EvaluationMethod object

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

## CheckReturnTypeEvaluationReal()

Checks if the Command Factory properly returns EvaluationReal object

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

## CheckReturnTypeFilledCircle()

Checks if the Command Factory properly returns Filled Circle object

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

## CheckReturnTypeFilledRectangle()

Checks if the Command Factory properly returns Filled Rectangle object

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

## CheckReturnTypeRectangle()

Checks if the Command Factory properly returns Rectangle object

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

## CheckReturnTypeTriangle()

Checks if the Command Factory properly returns Triangle object

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

## CheckReturnTypeWrite()

Checks if the Command Factory properly returns Write object

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

# Class AppParserTests

Namespace: [ASE Assignment Tests.ComponentTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the AppParser Class. Tests the returned objects and program parsing.

```
[TestClass]  
public class AppParserTests
```

Inheritance

[object](#) ← AppParserTests

Inherited Members

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

## Methods

### CheckBadCommand()

Checks if the Parser properly throws exception on illegal commands

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

### CheckMultiLineProgram()

Checks if the Parser properly parses a multi-line program

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

### CheckReturnTypeBrush()

Checks if the Parser properly returns Brush object

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

## CheckReturnTypeCircle()

Checks if the Parser properly returns circle object

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

## CheckReturnTypeEvaluationArray()

Checks if the Parser properly returns EvaluationArray object

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

## CheckReturnTypeEvaluationCall()

Checks if the Parser properly returns EvaluationCall object

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

## CheckReturnTypeEvaluationElse()

Checks if the Parser properly returns EvaluationElse object

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

## CheckReturnTypeEvaluationFor()

Checks if the Parser properly returns EvaluationFor object

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

## CheckReturnTypeEvaluationIf()

Checks if the Parser properly returns EvaluationIf object

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

## CheckReturnTypeEvaluationInt()

Checks if the Parser properly returns EvaluationInt object

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

## CheckReturnTypeEvaluationMethod()

Checks if the Parser properly returns EvaluationMethod object

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

## CheckReturnTypeEvaluationReal()

Checks if the Parser properly returns EvaluationReal object

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

## CheckReturnTypeFilledCircle()

Checks if the Parser properly returns filled circle object

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

## CheckReturnTypeFilledRectangle()

Checks if the Parser properly returns filled rectangle object

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

## CheckReturnTypeRectangle()

Checks if the Parser properly returns rectangle object

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

## CheckReturnTypeTriangle()

Checks if the Parser properly returns Triangle object

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

## CheckReturnTypeWrite()

Checks if the Parser properly returns Write object

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

## CheckSyntaxError()

Checks if the Parser properly throws exception on syntax errors

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

# Class AppStoredProgramTests

Namespace: [ASE Assignment Tests.ComponentTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the AppStoredProgram Class. Checks the exceptions from the class.

```
[TestClass]
public class AppStoredProgramTests
```

## Inheritance

[object](#) ← AppStoredProgramTests

## Inherited Members

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

## Methods

### PushAndPopConditionalCommand()

Checks if the push and pop can properly work for conditional commands

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

### ResetProgramRunsWithoutException()

Checks if the program can reset without exception

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

### RunProgramRunsWithoutException()

Checks if an empty program runs without exception

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

## TestAddAndRemoveProgram()

Checks if the program can add and remove command programs

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

## TestAddMethod()

Checks if methods can be added to the program

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

# Namespace ASE\_Assignment\_Tests.Evaluation Tests

## Classes

### [BooseWrapperTests](#)

The Unit tests for the Boose Wrapper class.

### [EvaluationCallTests](#)

The Unit tests for the EvaluationCall Class

# Class BoozeWrapperTests

Namespace: [ASE Assignment Tests.EvaluationTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the Booze Wrapper class.

```
[TestClass]
public class BoozeWrapperTests
```

Inheritance

[object](#) ← BoozeWrapperTests

Inherited Members

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

## Methods

### TestArrayEvaluation()

Tests the array evaluation command creation

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

### TestBrush()

Tests the brush command creation

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

### TestCircle()

Tests the circle command creation

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

## TestDrawToMainProgram()

Tests the drawto command passed through the parser

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

## TestElseEvaluation()

Tests the else evaluation command creation

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

## TestFilledCircle()

Tests the filled circle command creation

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

## TestFilledRectangle()

Tests the filled rectangle command creation

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

## TestForEvaluation()

Tests the for evaluation command creation

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

## TestIfEvaluation()

Tests the if evaluation command creation

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

## TestIntEvaluation()

Tests the int evaluation command creation

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

## TestMethodEvaluation()

Tests the method evaluation command creation

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

## TestMoveToMainProgram()

Tests the moveto command passed through the parser

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

## TestMultiLineMainProgram()

Tests system by passing in a multiline program

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

## TestRealEvaluation()

Tests the real evaluation command creation

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

## TestRectangle()

Tests the rectangle command creation

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

## TestTriangle()

Tests the triangle command creation

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

## TestWrite()

Tests the write command creation

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

# Class EvaluationCallTests

Namespace: [ASE Assignment Tests.EvaluationTests](#)

Assembly: ASE Assignment Tests.dll

The Unit tests for the EvaluationCall Class

```
[TestClass]  
public class EvaluationCallTests
```

Inheritance

[object](#) ← EvaluationCallTests

Inherited Members

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

## Methods

### CheckObjectCreation()

Checks if the class objects can be initialized

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

### CheckObjectCreationRestriction()

Checks if the restrictions are overridden by initializing 1000 objects

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

### CheckParameters\_InvalidParameterCount\_ThrowsException()

Checks if wrong parameter count throws exception

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

# Namespace BoozeWebApp.Server

## Classes

### [WeatherForecast](#)

Default generated when starting this project by visual studio. Gave errors while removing so had to keep it in for some reason.

# Class WeatherForecast

Namespace: [BooseWebApp.Server](#)

Assembly: BooseWebApp.Server.dll

Default generated when starting this project by visual studio. Gave errors while removing so had to keep it in for some reason.

```
public class WeatherForecast
```

Inheritance

[object](#) ← WeatherForecast

Inherited Members

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

## Properties

Date

```
public DateOnly Date { get; set; }
```

Property Value

[DateOnly](#)

Summary

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

Property Value

[string](#)

## TemperatureC

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

Property Value

[int ↗](#)

## TemperatureF

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

Property Value

[int ↗](#)

# Namespace BoozeWebApp.Server.Controllers

## Classes

### [BoozeController](#)

REST backend for running the booze interpreter

# Class BooseController

Namespace: [BooseWebApp.Server.Controllers](#)

Assembly: BooseWebApp.Server.dll

REST backend for running the boose interpreter

```
[ApiController]
[Route("[controller]")]
public class BooseController : ControllerBase
```

## Inheritance

[object](#) ← [ControllerBase](#) ← BooseController

## Inherited Members

[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\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [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.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

**BooseController(ILocator<BooseController>, BooseWrapper, IBooseProgramRepository)**

Public constructor to initialize the boose interpreter and repository

```
public BooseController(ILocator<BooseController> logger, BooseWrapper booseInterpreter,  
IBooseProgramRepository booseProgramRepository)
```

## Parameters

logger  [ILocator](#)< [BooseController](#)>

booseInterpreter  [BooseWrapper](#)

booseProgramRepository  [IBooseProgramRepository](#)

# Methods

## DeleteProgram(int)

Delete a saved program by its program ID

```
[HttpDelete("DeleteProgram/{program_id}")]
public Task<IActionResult> DeleteProgram(int program_id)
```

Parameters

program\_id [int](#)

Returns

[Task](#) <[IActionResult](#)>

Success or failed message

## GetPrograms()

Get request to get all the saved programs from the repository

```
[HttpGet("GetPrograms")]
public Task<IActionResult> GetPrograms()
```

Returns

[Task](#) <[IActionResult](#)>

Array of program objects in JSON format

## RunCommand(CommandInput)

Post request to run commands and return the output image and message

```
[HttpPost("RunCommand")]
public IActionResult RunCommand(CommandInput input)
```

Parameters

**input** [CommandInput](#)

Returns

[IActionResult](#)

JSON with base64 encoded image and message

## SaveProgram(BooseProgram)

Post request to save the program to the database passed in as JSON

```
[HttpPost("SaveProgram")]
public Task<IActionResult> SaveProgram(BooseProgram program)
```

Parameters

**program** [BooseProgram](#)

Returns

[Task](#) <[IActionResult](#)>

The saved program JSON

# Namespace BoozeWebApp.Server.Models

## Classes

### [BoozeProgram](#)

Entity to store booze programs in the database

### [CommandInput](#)

Model to accept commands

### [CommandOutput](#)

Model to send JSON output by the REST API

## Interfaces

### [IBoozeProgramRepository](#)

Interface for the Booze Programs Repository

# Class BooseProgram

Namespace: [BooseWebApp.Server.Models](#)

Assembly: BooseWebApp.Server.dll

Entity to store boose programs in the database

```
[Table("BooseProgram")]
public class BooseProgram
```

Inheritance

[object](#) ← BooseProgram

Inherited Members

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

## Properties

### Program\_Code

```
[MaxLength(4000)]
public string Program_Code { get; set; }
```

Property Value

[string](#)

### Program\_Name

```
[NotNull]
[MaxLength(100)]
public string Program_Name { get; set; }
```

Property Value

[string](#) ↗

## Program\_id

```
[PrimaryKey]  
[AutoIncrement]  
public int Program_id { get; set; }
```

Property Value

[int](#) ↗

# Class CommandInput

Namespace: [BooseWebApp.Server.Models](#)

Assembly: BooseWebApp.Server.dll

Model to accept commands

```
public class CommandInput
```

Inheritance

[object](#) ← CommandInput

Inherited Members

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

## Properties

Command

```
public string Command { get; set; }
```

Property Value

[string](#)

# Class CommandOutput

Namespace: [BooseWebApp.Server.Models](#)

Assembly: BooseWebApp.Server.dll

Model to send JSON output by the REST API

```
public class CommandOutput
```

## Inheritance

[object](#) ← CommandOutput

## Inherited Members

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

## Properties

### ImageBase64

```
public string ImageBase64 { get; set; }
```

Property Value

[string](#)

### Message

```
public string Message { get; set; }
```

Property Value

[string](#)

# Interface IBooseProgramRepository

Namespace: [BooseWebApp.Server.Models](#)

Assembly: BooseWebApp.Server.dll

Interface for the Boose Programs Repository

```
public interface IBooseProgramRepository
```

## Methods

AddAsync(BooseProgram)

```
Task AddAsync(BooseProgram user)
```

Parameters

`user` [BooseProgram](#)

Returns

[Task](#)

DeleteAsync(int)

```
Task DeleteAsync(int id)
```

Parameters

`id` [int](#)

Returns

[Task](#)

## GetAllAsync()

Task<List<BooseProgram>> GetAllAsync()

Returns

[Task](#) <[List](#) <[BooseProgram](#)>>

## GetByIdAsync(int)

Task<BooseProgram> GetByIdAsync([int](#) id)

Parameters

[id](#) [int](#)

Returns

[Task](#) <[BooseProgram](#)>

## GetByProgramNameAsync(string)

Task<BooseProgram> GetByProgramNameAsync([string](#) programName)

Parameters

[programName](#) [string](#)

Returns

[Task](#) <[BooseProgram](#)>

## UpdateAsync(BooseProgram)

Task [UpdateAsync](#)(BooseProgram user)

Parameters

user [BooseProgram](#)

Returns

[Task](#) ↗

# Namespace BoozeWebApp.Server.Repositories

## Classes

### [BoozeProgramRepository](#)

Interface for the Booze Programs Repository

# Class BoozeProgramRepository

Namespace: [BoozeWebApp.Server.Repositories](#)

Assembly: BoozeWebApp.Server.dll

Interface for the Booze Programs Repository

```
public class BoozeProgramRepository : IBoozeProgramRepository
```

Inheritance

[object](#) ← BoozeProgramRepository

Implements

[IBoozeProgramRepository](#)

Inherited Members

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

## Constructors

### BoozeProgramRepository()

Public Constructor to initialize the db and make the tables

```
public BoozeProgramRepository()
```

## Methods

### AddAsync(BoozeProgram)

Adds the program to the database

```
public Task AddAsync(BoozeProgram program)
```

Parameters

program [BooseProgram](#)

Returns

[Task](#)

The added program

## DeleteAsync(int)

Deletes the program specified by id

```
public Task DeleteAsync(int id)
```

Parameters

[id int](#)

Program id

Returns

[Task](#)

## GetAllAsync()

Gets all saved programs

```
public Task<List<BooseProgram>> GetAllAsync()
```

Returns

[Task](#) <[List](#) <[BooseProgram](#)>>

## GetByIdAsync(int)

Gets program by id

```
public Task<BooseProgram> GetByIdAsync(int id)
```

Parameters

**id** [int](#)

Returns

[Task](#) <[BooseProgram](#)>

A program with the id

## GetByProgramNameAsync(string)

```
public Task<BooseProgram> GetByProgramNameAsync(string programName)
```

Parameters

**programName** [string](#)

Returns

[Task](#) <[BooseProgram](#)>

## UpdateAsync(BooseProgram)

Updates the program specified

```
public Task UpdateAsync(BooseProgram program)
```

Parameters

**program** [BooseProgram](#)

Returns

[Task](#)

