

# Namespace Test\_BOOSE

## Classes

### [AppCanvasTest](#)

Unit test for AppCanvas class.

# Class AppCanvasTest

Namespace: [Test\\_BOOSE](#)

Assembly: Test\_BOOSE.dll

Unit test for AppCanvas class.

```
[TestClass]  
public class AppCanvasTest
```

Inheritance

[object](#) ← AppCanvasTest

Inherited Members

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

## Methods

### Circle\_Draw\_InvalidRadius()

Ensures that the Circle command throws CanvasException for invalid radius.

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]  
public void Circle_Draw_InvalidRadius()
```

### Circle\_Draw\_ValidRadius()

Verifies the Circle command to draw with valid radius.

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

### Clear\_CanvasContent()

Verifies that the Clear method clears the canvas drawing.

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

## DrawTo\_ThrowsExceptionForInvalidDestination()

Ensures that the DrawTo Method throws CanvasException for invalid destination position.

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]  
public void DrawTo_ThrowsExceptionForInvalidDestination()
```

## DrawTo\_UpdatesPenPositionCorrectly()

Verifies the DrawTo method updates the pen position correctly.

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

## MoveTo\_ThrowsExceptionForInvalidPosition()

Ensures that the MoveTo method throws Canvas Exception for invalid position.

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]  
public void MoveTo_ThrowsExceptionForInvalidPosition()
```

## MoveTo\_UpdatesPenPosition()

Verifies that the MoveTo methods updates the pen position correctly.

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

## Rectangle\_Draw\_InvalidDimensions()

Ensures that the Rectangle command throws CanvasException for invalid dimensions.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void Rectangle_Draw_InvalidDimensions()
```

## Rectangle\_Draw\_ValidDimensions()

Verifies the Rectangle command to draw with valid dimensions.

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

## Reset\_SetPenToOriginPosition()

Verifies that the Reset method reset the pen position to its origin.

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

## SetColour\_ThrowsException\_InvalidRGB()

Ensures SetColour throws a CanvasException for invalid RGB values.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void SetColour_ThrowsException_InvalidRGB()
```

## SetColourUpdatesPenColour\_ValidRGB()

Ensures the SetColour method updates the pen color correctly with valid RGB values.

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

## Set\_UpdatesCanvasSize()

Verifies the Set command updates canvas dimensions correctly and resets the pen position.

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

## TestMultilineProgram\_CommandsExecuteCorrectly()

Ensures multiple commands on the canvas and verifies the result.

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

## Triangle\_Draw\_InvalidDimensions()

Ensures that the Triangle command throws CanvasException for invalid dimensions.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void Triangle_Draw_InvalidDimensions()
```

## Triangle\_Draw\_ValidDimensions()

Verifies the Triangle command to draw with valid dimensions.

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

## WriteText\_EmptyText\_ThrowException()

Ensures that the Write command throws CanvasException for empty text.

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]  
public void WriteText_EmptyText_ThrowException()
```

## Write\_Draw\_ValidText()

Verifies the Write command to draw valid text.

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

# Namespace Test\_BOOSE.CommandsTest

## Classes

### [AppCallTest](#)

Unit test class for testing the functionality of the AppCall class.

### [AppElseTest](#)

Unit test class for testing the functionality of the AppElse class.

### [AppEndTests](#)

Contains unit tests for the AppEnd class.

### [AppIfTests](#)

Unit test class for testing the functionality of the AppIf class.

### [AppMethodTest](#)

Unit test class for testing the functionality of the AppMethod class.

### [AppWhileTest](#)

Contains unit tests for the AppWhile class.

### [CircleAppTest](#)

Unit test class for testing the functionality of the CircleApp class.

### [RectangleAppTest](#)

Contains unit tests for the RectangleApp class.

### [TriangleAppTest](#)

Contains unit tests for the TriangleApp class.

# Class AppCallTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppCall class.

```
[TestClass]  
public class AppCallTest
```

Inheritance

[object](#) ← AppCallTest

Inherited Members

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

## Methods

### TestConstructor\_CallsReduceRestrictions()

Test to verify that the constructor of AppCall calls the ReduceRestriction method. This test checks whether the ReduceRestrictions method is invoked during object initialization.

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

### TestRestrictions\_MethodDoesNotThrowException()

Test to ensure that the [Restrictions\(\)](#) method does not throw any exceptions. This is important because the method is not currently implemented and should not throw any exceptions.

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

# Class AppElseTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppElse class.

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

Inheritance

[object](#) ← AppElseTest

Inherited Members

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

## Methods

### Restrictions\_ShouldNotThrowException()

Test to verify that the Restriction method does not throw an exception. This test ensures that calling the Restrictions method does not result in any exceptions, as it is not currently implemented in the AppElse class.

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

# Class AppEndTests

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Contains unit tests for the AppEnd class.

```
[TestClass]
public class AppEndTests
```

Inheritance

[object](#) ← AppEndTests

Inherited Members

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

## Methods

### Restrictions\_NoRestrictionsDefined\_NoExceptionThrown()

Tests the behavior of the Restriction method when no restrictions are defined in the method. The test ensures that no exception is thrown when calling this method.

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

# Class AppIfTests

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppIf class.

```
[TestClass]  
public class AppIfTests
```

Inheritance

[object](#) ← AppIfTests

Inherited Members

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

## Methods

### Constructor\_ShouldCreateInstance()

Test to verify that an instance of the AppIf class can be successfully created.

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

### Restrictions\_ShouldNotThrowException()

Test to verify that the Restrictions method does not throw any exceptions when called. This test ensures that calling the Restrictions method does not result in any exceptions, as the method is not currently implemented in the AppIf class.

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

# Class AppMethodTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppMethod class.

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

Inheritance

[object](#) ← AppMethodTest

Inherited Members

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

## Methods

### TestConstructor\_ReducesRestrictions()

Test to verify that the constructor of the AppMethod class correctly reduces restrictions related to method count.

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

# Class AppWhileTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Contains unit tests for the AppWhile class.

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

Inheritance

[object](#) ← AppWhileTest

Inherited Members

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

## Methods

### Restrictions\_ShouldNotThrowAnyException\_WhenCalled()

Test to check that the Restriction method in the AppWhile class can be called without throwing any exceptions.

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

# Class CircleAppTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the CircleApp class.

```
[TestClass]  
public class CircleAppTest
```

Inheritance

[object](#) ← CircleAppTest

Inherited Members

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

## Methods

### CheckParameters\_ShouldSetRadius\_WhenParametersAreValid()

Validates that CheckParameters sets the radius correctly for valid parameters.

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

### Constructor\_Default\_ShouldCreateInstance()

Test to verify that the default constructor creates a valid instance of CircleApp class.

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

### Constructor\_Parameterized\_ShouldSetRadius()

Validates that the parameterized constructor initializes the radius correctly.

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

# Class RectangleAppTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Contains unit tests for the RectangleApp class.

```
[TestClass]
public class RectangleAppTest
```

Inheritance

[object](#) ← RectangleAppTest

Inherited Members

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

## Methods

### CheckParameters\_ShouldAcceptValidParameters()

Tests that the checkparameter method accepts valid parameters and validates them correctly.

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

### Constructor\_Default\_ShouldCreateInstance()

Tests that the default constructor of RectangleApp creates a new instance.

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

### Constructor\_Parameterized\_ShouldSetDimensions()

Tests that the parameterized constructor of RectangleApp initializes width and height correctly.

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

# Class TriangleAppTest

Namespace: [Test\\_BOOSE.CommandsTest](#)

Assembly: Test\_BOOSE.dll

Contains unit tests for the TriangleApp class.

```
[TestClass]
public class TriangleAppTest
```

Inheritance

[object](#) ← TriangleAppTest

Inherited Members

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

## Methods

### CheckParameters\_ShouldThrowCommandException\_WhenDimensionsAreNonPositive()

Tests that TriangleApp checkparameter method throws a commandexception when dimensions are non-positive.

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

### CheckParameters\_ShouldThrowCommandException\_WhenParameterCountIsIncorrect()

Tests that TriangleApp checkparameter method throws a commandexception when the number of parameters is incorrect.

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

## CheckParameters\_ShouldThrowCommandException\_WhenParametersAreNotValidIntegers()

Tests that TriangleApp checkparameter method throws a commandexception when parameters are not valid integers.

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

## CheckParameters\_ShouldThrowCommandException\_WhenParametersAreNull()

Tests that TriangleApp checkparameter methods that throws a commandexception when parameters are null.

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

## Constructor\_Default\_ShouldCreateInstance()

Tests that the default constructor of TriangleApp class creates a new instance.

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

## Constructor\_Parameterized\_ShouldSetDimensions()

Tests that the parameterized constructor of TriangleApp initializes width and height correctly.

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

## Execute\_ShouldDrawTriangle\_WhenParametersAreValid()

Tests that TriangleApp execute method calls the tri(int,int) method with correct dimensions.

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

# Namespace Test\_BOOSE.ComponentTest

## Classes

### [AppArrayTest](#)

Unit test class for testing the functionality of the AppArray class.

### [AppBooleanTest](#)

Unit test class for testing the functionality of the AppBoolean class.

### [AppIntTest](#)

Unit test class for testing the functionality of the AppInt class.

### [AppRealTest](#)

Unit test class for testing the functionality of the AppReal class.

# Class AppArrayTest

Namespace: [Test\\_BOOSE.ComponentTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppArray class.

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

Inheritance

[object](#) ← AppArrayTest

Inherited Members

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

## Methods

### Constructor\_ShouldCreateInstance()

Test to verify that an instance of AppArray class can be successfully created. This ensures that the constructor initializes the object correctly.

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

# Class AppBooleanTest

Namespace: [Test\\_BOOSE.ComponentTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppBoolean class.

```
[TestClass]
public class AppBooleanTest
```

Inheritance

[object](#) ← AppBooleanTest

Inherited Members

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

## Methods

### TestRestrictions\_MethodDoesNotThrowException()

Test to ensure that the Restriction method does not throw any exceptions. This is important because the method is not currently implemented, and no restrictions are applied.

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

# Class AppIntTest

Namespace: [Test\\_BOOSE.ComponentTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppInt class.

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

Inheritance

[object](#) ← AppIntTest

Inherited Members

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

## Methods

### AppInt\_Restrictions\_ShouldNotThrowException()

Test to verify that the creation of an AppInt class instance does not throw any exceptions and the instance is initialized properly.

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

### AppInt\_ShouldInitializeWithoutException()

Test to verify that an instance of AppInt class can be successfully created without any exceptions.

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

# Class AppRealTest

Namespace: [Test\\_BOOSE.ComponentTest](#)

Assembly: Test\_BOOSE.dll

Unit test class for testing the functionality of the AppReal class.

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

Inheritance

[object](#) ← AppRealTest

Inherited Members

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

## Methods

### Restrictions\_ShouldNotThrowAnyException\_WhenCalled()

Test to verify that the Restriction method in the AppReal class does not throw any exception when called.

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

# Namespace \_77356855\_Aastha\_Tamrakar ASE Assignment

## Classes

### [AppCanvas](#)

Represents a canvas for drawing and managing graphical elements.

### [AppCommandFactory](#)

Application-specific commands are created by this factory class. Extends the base CommandFactory to support custom application-specific commands.

### [Form1](#)

Represents the main form of BOOSE application which provide UI and functionality for the canvas and commands.

# Class AppCanvas

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a canvas for drawing and managing graphical elements.

```
public class AppCanvas : ICanvas
```

## Inheritance

[object](#) ← AppCanvas

## Implements

ICanvas

## Inherited Members

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

# Constructors

## AppCanvas()

Initializes a new instance for AppCanvas with default size.

```
public AppCanvas()
```

# Properties

## PenColour

Gets or sets the current pen color.

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

## Property Value

## Xpos

Gets or sets the current X position of the pen.

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

### Property Value

[int](#)

### Exceptions

CanvasException

Thrown when the position is beyond the boundaries of the canvas. See BOOSE.CanvasException for details.

## Ypos

Gets or sets the current Y position of the pen.

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

### Property Value

[int](#)

### Exceptions

CanvasException

Thrown when the position is beyond the boundaries of the canvas. See BOOSE.CanvasException for details.

## Methods

## Circle(int, bool)

Draws circle at current pen position.

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

### Parameters

**radius** [int](#)

The radius of the circle.

**filled** [bool](#)

Indicates whether the circle should be filled.

### Exceptions

CanvasException

Thrown when the radius is negative or the graphics object is not initialized. See BOOSE.CanvasException for details.

## Clear()

Clears the canvas by filling it with default background color.

```
public void Clear()
```

### Exceptions

CanvasException

Thrown when the graphic object is not initialized. See BOOSE.CanvasException for details.

## DrawTo(int, int)

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

```
public void DrawTo(int toX, int toY)
```

## Parameters

**toX** [int](#)

The target X position.

**toY** [int](#)

The target Y position.

## Exceptions

CanvasException

Thrown when the position is beyond the boundaries of the canvas or the graphics object is not initialized. See BOOSE.CanvasException for details.

## MoveTo(int, int)

Moves the pen to the specified position without drawing.

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

## Parameters

**x** [int](#)

The target X position.

**y** [int](#)

The target Y position.

## Exceptions

CanvasException

Thrown when the position is beyond the boundaries of the canvas. See BOOSE.CanvasException for details.

## Rect(int, int, bool)

Draws a rectangle at the current pen position.

```
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](#)

Indicates whether the rectangle should be filled.

### Exceptions

[CanvasException](#)

Thrown when the dimensions are negative or the graphics object is not initialized. See [BOOSE.CanvasException](#) for details.

## Reset()

Resets the pen position to the origin (0, 0).

```
public void Reset()
```

## Set(int, int)

Sets the size of the canvas and resets the pen position.

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

## Parameters

**xsize** [int](#)

The width of the canvas.

**ysize** [int](#)

The height of the canvas.

## SetColour(int, int, int)

Set the pen color using RGB values.

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

## Parameters

**red** [int](#)

The red component (0-255).

**green** [int](#)

The green component (0-255).

**blue** [int](#)

The blue component (0-255).

## Exceptions

CanvasException

Thrown when the color value is out of range. See BOOSE.CanvasException for details.

## Tri(int, int)

Draws a triangle at the current pen position.

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

## Parameters

**width** [int](#)

The width of the triangle.

**height** [int](#)

The height of the triangle.

## Exceptions

CanvasException

Thrown when the dimensions are negative or the graphics object is not initialized. See BOOSE.CanvasException for details.

## WriteText(string)

Writes text at the current pen position.

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

## Parameters

**text** [string](#)

The text to write.

## Exceptions

CanvasException

Thrown when the text is null, empty, or the graphics object is not initialized. See BOOSE.CanvasException for details.

## getBitmap()

Get the bitmap representation of canavs.

```
public object getBitmap()
```

Returns

object ↗

The current bitmap.

# Class AppCommandFactory

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment](#)

Assembly: 77356855\_Aastha\_Tamrakar\_ASE\_Assignment.dll

Application-specific commands are created by this factory class. Extends the base CommandFactory to support custom application-specific 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\(\)](#)

## Constructors

### AppCommandFactory()

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

```
public AppCommandFactory()
```

## Methods

### MakeCommand(string)

Creates a command instance based on the specified command type. This factory supports application-specific commands such as "tri", "write", and others.

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

## Parameters

### `commandType` [string](#) ↗

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

## Returns

### ICommand

An instance of the corresponding command based on the `commandType`. If no specific command matches, delegates to the base class to create default commands.

## Remarks

The `commandType` string is converted to lowercase and trimmed before matching. Example command types include "tri" for drawing a triangle, "write" for writing functionality, and various others for drawing shapes, conditional logic, loops, etc.

# Class Form1

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment](#)

Assembly: 77356855\_Aastha\_Tamrakar\_ASE\_Assignment.dll

Represents the main form of BOOSE application which provide UI and functionality for the canvas and commands.

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

## Inheritance

```
object ↗ ← MarshalByRefObject ↗ ← Component ↗ ← Control ↗ ← ScrollableControl ↗ ←  
ContainerControl ↗ ← Form ↗ ← Form1
```

## Implements

```
IDropTarget ↗ , ISynchronizeInvoke ↗ , IWin32Window ↗ , IBindableComponent ↗ , IComponent ↗ ,  
IDisposable ↗ , IContainerControl ↗
```

## Inherited Members

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

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

[ContainerControl.CurrentAutoScaleDimensions](#) , [ContainerControl.ParentForm](#) ,  
[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,  
[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,  
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,  
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,  
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,  
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,  
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,  
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,  
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,  
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,  
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,  
[Control.BeginInvoke\(Action\)](#) , [Control.BeginInvoke\(Delegate, params object\[\]\)](#) ,  
[Control.BringToFront\(\)](#) , [Control.Contains\(Control\)](#) , [Control.CreateGraphics\(\)](#) ,  
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [Control.DoDragDrop\(object, DragDropEffects\)](#) ,  
[Control.DoDragDrop\(object, DragDropEffects, Bitmap, Point, bool\)](#) ,  
[Control.DrawToBitmap\(Bitmap, Rectangle\)](#) , [Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) ,  
[Control.GetTopLevel\(\)](#) , [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) ,  
[Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) , [Control.Focus\(\)](#) ,  
[Control.FromChildHandle\(nint\)](#) , [Control.FromHandle\(nint\)](#) ,  
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,  
[Control.GetContainerControl\(\)](#) , [Control.GetNextControl\(Control, bool\)](#) ,  
[Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) , [Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) ,  
[Control.Invalidate\(Region, bool\)](#) , [Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) ,  
[Control.Invalidate\(Rectangle\)](#) , [Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Action\)](#) ,  
[Control.Invoke\(Delegate\)](#) , [Control.Invoke\(Delegate, params object\[\]\)](#) ,  
[Control.Invoke<T>\(Func<T>\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,  
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,  
[Control.IsAnyInputChar\(char\)](#) , [Control.IsAnyInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,  
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,  
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,  
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.OnBindingContextChanged\(EventArgs\)](#) ,  
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,

[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,  
[Control.OnDockChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,  
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,  
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,  
[Control.OnParentDataContextChanged\(EventArgs\)](#) , [Control.OnParentEnabledChanged\(EventArgs\)](#) ,  
[Control.OnParentFontChanged\(EventArgs\)](#) , [Control.OnParentForeColorChanged\(EventArgs\)](#) ,  
[Control.OnParentRightToLeftChanged\(EventArgs\)](#) , [Control.OnParentVisibleChanged\(EventArgs\)](#) ,  
[Control.OnPrint\(PaintEventArgs\)](#) , [Control.OnTabIndexChanged\(EventArgs\)](#) ,  
[Control.OnTabStopChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnLocationChanged\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.InvokeGotFocus\(Control, EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidate\(InvalidateEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyPressEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClicked\(MouseEventArgs\)](#) ,  
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseDown\(MouseEventArgs\)](#) ,  
[Control.OnMouseEnter\(EventArgs\)](#) , [Control.OnMouseLeave\(EventArgs\)](#) ,  
[Control.OnDpiChangedBeforeParent\(EventArgs\)](#) , [Control.OnDpiChangedAfterParent\(EventArgs\)](#) ,  
[Control.OnMouseHover\(EventArgs\)](#) , [Control.OnMouseMove\(MouseEventArgs\)](#) ,  
[Control.OnMouseUp\(MouseEventArgs\)](#) ,  
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnSystemColorsChanged\(EventArgs\)](#) , [Control.OnValidating\(CancelEventArgs\)](#) ,  
[Control.OnValidated\(EventArgs\)](#) , [Control.PerformLayout\(\)](#) , [Control.PerformLayout\(Control, string\)](#) ,  
[Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[Control.RaiseDragEvent\(object, DragEventArgs\)](#) , [Control.RaisePaintEvent\(object, PaintEventArgs\)](#) ,  
[Control.RecreateHandle\(\)](#) , [Control.RectangleToClient\(Rectangle\)](#) ,  
[Control.RectangleToScreen\(Rectangle\)](#) , [Control.ReflectMessage\(nint, ref Message\)](#) ,  
[Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) , [Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) ,  
[Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) , [Control.Select\(\)](#) ,  
[Control.SelectNextControl\(Control, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,

[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,  
[Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,  
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,  
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,  
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,  
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.DataContext](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.CompanyName](#) ,  
[Control.ContainsFocus](#) , [Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) ,  
[Control.Cursor](#) , [Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) ,  
[Control.DefaultFont](#) , [Control.DefaultForeColor](#) , [Control.DefaultMargin](#) ,  
[Control.DefaultMaximumSize](#) , [Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) ,  
[Control.DeviceDpi](#) , [Control.IsDisposed](#) , [Control.Disposing](#) , [Control.Dock](#) ,  
[Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) , [Control.Font](#) ,  
[Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) , [Control.Height](#) ,  
[Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.Accessible](#) ,  
[Control.IsAncestorSiteInDesignMode](#) , [Control.IsMirrored](#) , [Control.Left](#) , [Control.Margin](#) ,  
[Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) , [Control.Name](#) ,  
[Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) , [Control.RecreatingHandle](#) ,  
[Control.Region](#) , [Control.RenderRightToLeft](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,  
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.TabIndex](#) , [Control.TabStop](#) ,  
[Control.Tag](#) , [Control.Top](#) , [Control.TopLevelControl](#) , [Control.ShowKeyboardCues](#) ,  
[Control.ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) , [Control.Width](#) ,  
[Control.PreferredSize](#) , [Control.Padding](#) , [Control.ImeMode](#) , [Control.ImeModeBase](#) ,  
[Control.PropagatingImeMode](#) , [Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,  
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,  
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) ,  
[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,  
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,  
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,  
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,

[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,  
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DataContextChanged](#) ,  
[Control.DragDrop](#) , [Control.DragEnter](#) , [Control.DragOver](#) , [Control.DragLeave](#) ,  
[Control.GiveFeedback](#) , [Control.HandleCreated](#) , [Control.HandleDestroyed](#) ,  
[Control.HelpRequested](#) , [Control.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

### Form1()

Initializes a new instance of form class. Sets up canvas, command factory, stored program, and parser.

```
public Form1()
```

## Methods

### Dispose(bool)

Clean up any resources being used.

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

## Parameters

### **disposing** bool ↗

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

# Namespace \_77356855\_Aastha\_Tamrakar\_ASE\_Assignment.Commands

## Classes

### [AppCall](#)

Represents a custom implementation of the BOOSE.Call class that reduces restrictions.

### [AppElse](#)

Represents the [AppElse](#) command in the application, which is part of a compound command structure. This class extends the base BOOSE.Else class and overrides its Restrictions method.

### [AppEnd](#)

Represents a custom implementation of the [AppEnd](#) class. This class overrides the Restrictions method from the base class.

### [AppFor](#)

Represents a custom implementation of the [AppFor](#) class. This class overrides the Restriction method from the base class.

### [AppIf](#)

Represents the [AppIf](#) class which inherits from the AppCompoundCommand class.

### [AppMethod](#)

Represents a custom implementation of the [AppMethod](#) class. This class overrides the constructor to reduce restrictions on method count.

### [AppWhile](#)

Represents a compound command that implements a "while" loop structure in the application. Inherits from the BOOSE.While class and provides a customizable implementation.

### [CircleApp](#)

Represents a command to draw a circle on a canvas.

### [ClearApp](#)

Represents a command to clear the canvas.

### [RectangleApp](#)

Represents a rectangle drawing command that requires two parameters: width and height.

### [ResetApp](#)

Represents a reset command to reset the canvas state.

### [TriangleApp](#)

Command that represents to draw triangle on the canvas. This command requires two parameters : width and height.

### [WriteApp](#)

Represents a command to write text on the canvas. The command accepts one parameter: the text to be written.

# Class AppCall

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a custom implementation of the BOOSE.Call class that reduces restrictions.

```
public class AppCall : Call, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

Call.methodName , Call.Compile() , Call.Execute() , 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

## AppCall()

Initializes a new instance of the [AppCall](#) class. Calls the Restrictions method during initialization to reduce any restrictions.

```
public AppCall()
```

## Methods

### Restrictions()

Overrides the Restrictions method. This method is intended to define specific restrictions but is currently not implemented.

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

# Class AppElse

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents the [AppElse](#) command in the application, which is part of a compound command structure. This class extends the base BOOSE.Else class and overrides its Restrictions method.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Methods

### Restrictions()

Overrides the Restrictions method. This method is intended to define any restrictions related to the Else command but is currently not implemented.

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

# Class AppEnd

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a custom implementation of the [AppEnd](#) class. This class overrides the Restrictions method from the base class.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Methods

### Restrictions()

Overrides the Restrictions method. This method is intended to define specific restrictions but is currently not implemented.

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

# Class AppFor

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a custom implementation of the [AppFor](#) class. This class overrides the Restriction method from the base class.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

# Methods

## Restrictions()

Overrides the Restriction method. This method is intended to define specific restrictions but is currently not implemented.

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



# Class AppIf

Namespace: [77356855\\_Aastha\\_Tamrakar\\_ASE\\_Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar\_ASE\_Assignment.dll

Represents the [AppIf](#) class which inherits from the AppCompoundCommand class.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Constructors

### AppIf()

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

```
public AppIf()
```

## Methods

### Restrictions()

A protected method intended to reduce or simplify restrictions. This method is currently not implemented.

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

# Class AppMethod

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a custom implementation of the [AppMethod](#) class. This class overrides the constructor to reduce restrictions on method count.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Constructors

### AppMethod()

Initializes a new instance of the [AppMethod](#) class. Calls the ReduceRestriction method to remove any restriction on method count.

```
public AppMethod()
```

# Class AppWhile

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a compound command that implements a "while" loop structure in the application. Inherits from the BOOSE.While class and provides a customizable implementation.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Constructors

### AppWhile()

Initializes a new instance of the [AppWhile](#) class. This constructor can be used to perform any setup necessary for the "while" loop implementation.

```
public AppWhile()
```

## Methods

### Restrictions()

Overrides the default restrictions for the "while" loop. This method can be used to define or customize restrictions specific to the "while" loop structure.

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

# Class CircleApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a command to draw a circle on a canvas.

```
public class CircleApp : CommandOneParameter, ICommand
```

## Inheritance

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

## 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

### CircleApp()

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

```
public CircleApp()
```

### CircleApp(Canvas, int)

Initializes a new instance of the [CircleApp](#) class with a specified canvas and radius.

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

## Parameters

**canvas** Canvas

The canvas where the circle will be drawn.

**radius** [int](#)

The radius of the circle.

## Methods

**CheckParameters(string[])**

Validates the parameters provided for the circle command.

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

## Parameters

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

An array of parameters where the first value should be the radius.

## Exceptions

**CommandException**

Thrown when the parameter list is invalid or the radius is not a valid integer.

[ArgumentOutOfRangeException](#)

Thrown when the radius is not a positive integer.

[Exception](#)

Thrown for any unexpected errors during parameter validation.

## Execute()

Executes the command to draw a circle with the specified radius on the canvas.

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

## Exceptions

RestrictionException

Thrown when the radius exceeds the maximum allowable value (2000).

[ArgumentException](#)

Thrown when a parameter is out of the valid range.

[InvalidOperationException](#)

Thrown when there is an error during execution.

[Exception](#)

Thrown for any unexpected errors during execution.

# Class ClearApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a command to clear the canvas.

```
public class ClearApp : CanvasCommand, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Methods

### CheckParameters(string[])

Validates the parameters for the clear command.

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

## Parameters

parameter [string](#)[]

An array of parameters to pass to the command.

## Remarks

This method checks if the `parameter` array has more than one element and throws an exception if it does.

## Exceptions

### [ArgumentException](#)

Thrown when the clear command is provided with parameters.

## Execute()

Executes the clear command to clear the canvas.

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

## Remarks

This method calls the Execute method and then clears the canvas by calling Clear class.

# Class RectangleApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a rectangle drawing command that requires two parameters: width and height.

```
public class RectangleApp : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

### RectangleApp()

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

```
public RectangleApp()
```

### RectangleApp(Canvas, int, int)

Initializes a new instance of the [RectangleApp](#) class with specified canvas, width, and height.

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

## Parameters

**canvas** Canvas

The canvas where the rectangle will be drawn.

**width** [int](#)

The width of the rectangle.

**height** [int](#)

The height of the rectangle.

## Methods

### CheckParameters(string[])

Checks the parameters provided for the rectangle command. Validates that exactly two positive integer parameters (width and height) are provided and within the allowed range.

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

## Parameters

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

The list of parameters as strings.

### Execute()

Executes the command to draw a rectangle on the canvas. Validates the width and height parameters and ensures they do not exceed the maximum allowed values.

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

# Class ResetApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a reset command to reset the canvas state.

```
public class ResetApp : CanvasCommand, ICommand
```

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Methods

### CheckParameters(string[])

Validates the parameters for the reset command.

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

## Parameters

parameter [string](#)[]

An array of parameters to pass to the command.

## Exceptions

### [ArgumentException](#)

Thrown when the reset command is provided with parameters.

## Execute()

Executes the reset command to reset the Canvas.

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

## Remarks

This method calls the Reset method to reset all the canvas drawings to their default state.

# Class TriangleApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Command that represents to draw triangle on the canvas. This command requires two parameters : width and height.

```
public class TriangleApp : CommandTwoParameters, ICommand
```

## Inheritance

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

## 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

### TriangleApp()

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

```
public TriangleApp()
```

## TriangleApp(Canvas, int, int)

Initializes a new instance of the TriangleApp class with specified canvas, width, and height.

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

### Parameters

**canvas** Canvas

The Canvas on which the triangle should be drawn.

**width** [int](#)

The width of the triangle.

**height** [int](#)

The height of the triangle.

## Methods

### CheckParameters(string[])

Checks and validates parameters for the triangle drawing command. Ensures only two parameters are provided.

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

### Parameters

**parameters** [string](#)[]

The parameters to validate.

### Exceptions

CommandException

Thrown when the number of parameters are incorrect or invalid.

## Execute()

Executes the triangle drawing command on the Canvas.

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

# Class WriteApp

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Commands](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a command to write text on the canvas. The command accepts one parameter: the text to be written.

```
public class WriteApp : CommandOneParameter, ICommand
```

## Inheritance

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

## 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

## WriteApp()

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

```
public WriteApp()
```

## WriteApp(Canvas, string)

Initializes a new instance of the WriteApp class with specified Canvas and text.

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

## Parameters

c Canvas

The Canvas where the text will be written.

text [string](#)

The text to write on the Canvas.

## Methods

### CheckParameters(string[])

Checks and validates the parameter for the text writing command. Ensures exactly one parameter is provided and is valid.

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

## Parameters

parameterList [string](#)[]

The parameters to validate.

## Exceptions

CommandException

Thrown when the parameter count is incorrect or the text is invalid.

## Execute()

Executes the text writing command on the Canvas.

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

## Exceptions

### RestrictionException

Thrown when the text is null, empty, whitespace, or exceeds 100 characters.

# Namespace \_77356855\_Aastha\_Tamrakar ASE Assignment.Components

## Classes

### [AppArray](#)

Represents an application-specific implementation of the BOOSE Array class. This class extends the functionality of the BOOSE Array to fit the specific requirements of the application.

### [AppBoolean](#)

Represents a custom implementation of the BOOSE.Boolean class. This class is used for handling boolean variables in the application.

### [AppInt](#)

Represents an integer type with custom restrictions for the application.

### [AppReal](#)

Represents the [AppReal](#) class which inherits from the BOOSE.Real class.

### [AppStoredProgram](#)

Represents a custom implementation of a stored program that runs commands on a specified canvas. This class overrides the [Run\(\)](#) method to process and execute commands until no more are left.

# Class AppArray

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Components](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents an application-specific implementation of the BOOSE Array class. This class extends the functionality of the BOOSE Array to fit the specific requirements of the application.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

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

## Constructors

### AppArray()

Initializes a new instance of the [AppArray](#) class. This constructor calls ReduceRestrictionCounter to adjust the restriction counter as part of the initialization process.

```
public AppArray()
```

# Class AppBoolean

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Components](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a custom implementation of the BOOSE.Boolean class. This class is used for handling boolean variables in the application.

```
public class AppBoolean : Boolean, ICommand
```

## Inheritance

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

## 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()

Specifies the restrictions or rules for the boolean values. This method can be overridden to define custom restrictions for boolean values.

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

# Class AppInt

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Components](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents an integer type with custom restrictions for the application.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

Int.Compile() , Int.Execute() , Evaluation.expression , Evaluation.evaluatedExpression ,  
Evaluation.varName , Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) ,  
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,  
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,  
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,  
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,  
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

# Constructors

## AppInt()

Initializes a new instance of the [AppInt](#) class. Calls the Restrictions method to apply any restrictions on the integer type.

```
public AppInt()
```

# Methods

## Restrictions()

Applies custom restrictions to the integer type. This method can be overridden to define specific restrictions.

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

# Class AppReal

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Components](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents the [AppReal](#) class which inherits from the BOOSE.Real class.

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

## Inheritance

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

## Implements

ICommand

## Inherited Members

Real.Compile() , Real.Execute() , Real.Value , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

# Methods

## Restrictions()

Overrides the Restrictions method from the base BOOSE.Real class. This method is intended to define specific restrictions but is currently not implemented.

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

# Class AppStoredProgram

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.Components](#)

Assembly: 77356855\_Aastha\_Tamrakar\_ASE\_Assignment.dll

Represents a custom implementation of a stored program that runs commands on a specified canvas. This class overrides the [Run\(\)](#) method to process and execute commands until no more are left.

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

## Inheritance

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

## Implements

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

## Inherited Members

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

[ArrayList.Reverse\(\)](#) , [ArrayList.Reverse\(int, int\)](#) , [ArrayList.SetRange\(int, ICollection\)](#) ,  
[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\(\)](#)

## Constructors

### AppStoredProgram(ICanvas)

Initializes a new instance of the AppStoredProgram class. This constructor initializes the stored program with a specified Canvas.

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

#### Parameters

**canvas** ICanvas

The Canvas on which the program will run its commands.

## Methods

### Run()

Executes the stored program by processing and executing all the commands sequentially. The method continues to execute commands as long as there are remaining commands.

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

#### Remarks

The method checks if there are commands left in the program using CommandsLeft. If there are commands, it retrieves the next command using NextCommand and executes it. The loop continues until there are no more commands to execute.

# Namespace \_77356855\_Aastha\_Tamrakar ASE Assignment.FileOperation

## Classes

### [LoadCanvas](#)

Represents a class that handles loading a canvas image from a file.

### [ReadCommand](#)

Represents a class responsible for reading commands from a text file.

### [SaveCanvas](#)

Represents a class responsible for saving the canvas image to a file.

### [WriteCommand](#)

Represents a class responsible for writing the given commands to a text file.

# Class LoadCanvas

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.FileOperation](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a class that handles loading a canvas image from a file.

```
public class LoadCanvas
```

## Inheritance

[object](#) ← LoadCanvas

## Inherited Members

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

## Constructors

### LoadCanvas()

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

```
public LoadCanvas()
```

## Methods

### Load()

Opens a file dialog to allow the user to select an image file and loads it into a bitmap.

```
public Bitmap Load()
```

Returns

[Bitmap](#)

A [Bitmap](#) object representing the loaded canvas image, or `null` if the operation was canceled or failed.

## Remarks

The supported image formats are PNG, JPG, and BMP. If an error occurs while loading the image, an error message will be displayed.

# Class ReadCommand

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.FileOperation](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a class responsible for reading commands from a text file.

```
public class ReadCommand
```

## Inheritance

[object](#) ← ReadCommand

## Inherited Members

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

# Constructors

## ReadCommand()

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

```
public ReadCommand()
```

# Methods

## Read()

Opens a file dialog to allow the user to select a text file and loads its content as a string.

```
public string Read()
```

Returns

[string](#)

A string containing the contents of the selected file. If the operation is canceled or fails, an empty string is returned.

## Remarks

The file dialog filters files to show only text files (with a .txt extension), but all file types can be selected. If an error occurs during file reading, an error message will be displayed to the user.

# Class SaveCanvas

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.FileOperation](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a class responsible for saving the canvas image to a file.

```
public class SaveCanvas
```

## Inheritance

[object](#) ← SaveCanvas

## Inherited Members

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

## Constructors

### SaveCanvas()

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

```
public SaveCanvas()
```

## Methods

### Save(Bitmap)

Opens a save file dialog to allow the user to choose a location and format to save the canvas as an image.

```
public void Save(Bitmap canvasBitmap)
```

## Parameters

canvasBitmap [Bitmap](#)

The [Bitmap](#) representing the current canvas to be saved.

## Remarks

The method supports saving the canvas in multiple formats, including PNG, JPEG, and BMP. The format is determined based on the file extension chosen by the user. If an error occurs during the save process, an error message will be displayed to the user.

# Class WriteCommand

Namespace: [77356855\\_Aastha\\_Tamrakar ASE Assignment.FileOperation](#)

Assembly: 77356855\_Aastha\_Tamrakar ASE Assignment.dll

Represents a class responsible for writing the given commands to a text file.

```
public class WriteCommand
```

## Inheritance

[object](#) ← WriteCommand

## Inherited Members

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

## Constructors

### WriteCommand()

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

```
public WriteCommand()
```

## Methods

### Write(string)

Opens a save file dialog and writes the given commands to the selected text file.

```
public void Write(string Commands)
```

## Parameters

Commands [string](#)

A string containing the commands to be saved to the file.

## Remarks

This method allows the user to choose the location and file name for saving the commands. If an error occurs during the saving process, an error message is shown.