

Class AcademicYear

This class defines a `AcademicYear` object.

Implements `IEnumerable` to allow Student objects inside the AcademicYear object to be iterated through. Implements `IComparable` to allow AcademicYear objects to be compared.

Author: Yashwant Rathor

Inheritance

- System.Object
 - AcademicYear

Implements

System.Collections.IEnumerable

System.IComparable<[AcademicYear](#)>

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [SchoolReportSystem.model](#)

Assembly: SchoolReportSystem.dll

Syntax

```
public class AcademicYear : IEnumerable, IComparable<AcademicYear>
```

🔗 Constructors

🔗 `AcademicYear(Int32)`

This custom constructor is responsible for creating an AcademicYear object.

Declaration

```
public AcademicYear(int year)
```

Parameters

Type	Name	Description
System.Int32	<i>year</i>	The school year number.

Examples

```
AcademicYear yr = new AcademicYear(7);
```

This creates an AcademicYear object with the year as '7'.

🔗 Properties

🔗 `GetStudents`

This method retrieves all students within the year.

Declaration

```
public List<Student> GetStudents { get; }
```

Property Value

Type	Description
System.Collections.Generic.List<Student>	returns the value of GetStudents.

Examples

```
AcademicYear yr = new AcademicYear(13);
yr.AddStudentToYear(new Student("8156", "Colin", "Jacobs", 13, "Lilac"));
yr.AddStudentToYear(new Student("8601", "Harris", "Ford", 13, "Rose"));
yr.AddStudentToYear(new Student("9509", "Scott", "Parker", 13, "Gold"));
yr.GetStudents;
```

This will return {("8156", "Colin", "Jacobs", 13, "Lilac"), ("8601", "Harris", "Ford", 13, "Rose"), ("9509", "Scott", "Parker", 13, "Gold")}.

[GetYearNo](#)

This method retrieves the year's number.

Declaration

```
public int GetYearNo { get; }
```

Property Value

Type	Description
System.Int32	returns the value of GetYearNo.

Examples

```
AcademicYear yr = new AcademicYear(12);
yr.GetYearNo;
```

This will return '12'.

[Methods](#)

[AddStudentToYear\(Student\)](#)

This method stores a valid Student object into the AcademicYear object.

Declaration

```
public void AddStudentToYear(Student s)
```

Parameters

Type	Name	Description
Student	s	Student object to be added to the year.

Examples

```
AcademicYear yr = new AcademicYear(12);
yr.AddStudentToYear(new Student("8126", "Edward", "Willis", 12, "Red"));
```

A new student has been added to 'yr'.

[CompareTo\(AcademicYear\)](#)

This method compares one AcademicYear object with another AcademicYear object.

Declaration

```
public int CompareTo(AcademicYear other)
```

Parameters

Type	Name	Description
AcademicYear	<i>other</i>	A different AcademicYear object.

Returns

Type	Description
System.Int32	returns 0 if the objects are the same and -1 or 1 if they are different.

Examples

```
AcademicYear yr = new AcademicYear(7);
AcademicYear yr2 = new AcademicYear(7);
yr.CompareTo(yr2);
```

This returns **0**, as the objects have the exact same year number.

[GetEnumerator\(\)](#)

This method returns the IEnumerator for the AcademicYear class.

Declaration

```
public IEnumerator GetEnumerator()
```

Returns

Type	Description
System.Collections.IEnumerator	returns the iteration of the non-generic collection for an AcademicYear object.

[GetStudent\(Int32\)](#)

This method retrieves the Student object located at a given index.

Declaration

```
public Student GetStudent(int index)
```

Parameters

Type	Name	Description
System.Int32	<i>index</i>	The index position of the desired Student object.

Returns

Type	Description
Student	returns the student located at the index position specified by the 'index' parameter.

Examples

```
AcademicYear yr = new AcademicYear(13);
yr.AddStudentToYear(new Student("8156", "Colin", "Jacobs", 13, "Lilac"));
yr.AddStudentToYear(new Student("8601", "Hamis", "Ford", 13, "Rose"));
yr.AddStudentToYear(new Student("9509", "Scott", "Parker", 13, "Gold"));
yr.GetStudent(2);
```

The student at index position 2 (Scott Parker) will be returned.

[GetStudentById\(String\)](#)

This method retrieves the Student object matched by the 'ID' string.

Declaration

```
public Student GetStudentById(string ID)
```

Parameters

Type	Name	Description
System.String	<i>ID</i>	The ID string to be used to get the matching Student object.

Returns

Type	Description
Student	returns the student with the matching ID specified by the 'ID' parameter.

Examples

```
AcademicYear yr = new AcademicYear(7);
yr.AddStudentToYear(new Student("0796", "Christine", "Lively", 7, "Indigo"));
yr.AddStudentToYear(new Student("1472", "Jessica", "Rhodes", 7, "Magenta"));
yr.AddStudentToYear(new Student("1140", "Frank", "London", 7, "Cyan"));
yr.GetStudentById("0796");
```

The student with the ID value '0796' (Christine Lively) will be returned.

[RemoveDuplicateStudents\(\)](#)

This method removes any duplicate students inside the AcademicYear object.

Declaration

```
public void RemoveDuplicateStudents()
```

Examples

```
AcademicYear yr = new AcademicYear(9);
yr.AddStudentToYear(new Student("2950", "Wendy", "Foakes", 9, "Maue"));
yr.AddStudentToYear(new Student("2950", "Wendy", "Foakes", 9, "Maue"));
yr.AddStudentToYear(new Student("4125", "Stewart", "Richards", 9, "Teal"));
yr.RemoveStudentFromYear(1);
```

The second occurrence of 'Wendy Foakes' will be removed as it already exists once.

[RemoveStudentFromYear\(Int32\)](#)

This method removes a student at a particular index position, inside the AcademicYear object.

Declaration

```
public void RemoveStudentFromYear(int i)
```

Parameters

Type	Name	Description
System.Int32	<i>i</i>	

Examples

```
AcademicYear yr = new AcademicYear(10);
yr.AddStudentToYear(new Student("5003", "Chris", "Archer", 10, "Green"));
yr.AddStudentToYear(new Student("4500", "Hayden", "Collins", 10, "Red"));
yr.AddStudentToYear(new Student("5210", "James", "Holder", 10, "Blue"));
yr.RemoveStudentFromYear(1);
```

The student at index position 1 (Hayden Collins) will be removed.

[ToString\(\)](#)

This method overrides the default 'ToString()' representation of the AcademicYear class.

Declaration

```
public override string ToString()
```

Returns

Type	Description
System.String	

Overrides

System.Object.ToString()

Examples

```
AcademicYear yr = new AcademicYear(10);  
yr.ToString();
```



Implements

System.Collections.IEnumerable

System.IComparable<T>