Show / Hide Table of Contents

Class Subject

This class defines a `Subject` object.

Implements IComparable to allow Subject objects to be compared.

Author: Yashwant Rathor

Inheritance

□ System.Object
  □ Subject

Implements

System.IComparable<Subject>

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.classes

Assembly: SchoolReportSystem.dll

Syntax

```
public class Subject : IComparable<Subject>
```

Constructors

Subject(String, String, String, String)

This custom constructor method is responsible for creating a Subject object.

Declaration

```
public Subject(string ID, string name, string actualGrade, string expectedGrade)
```

Parameters

Type	Name	Description
System.String	*ID*	The ID of the subject.
System.String	*name*	The name of the subject.
System.String	*actualGrade*	The actual subject grade.
System.String	*expectedGrade*	The expected subject grade.

Examples

```
Subject sub = new Subject("PHY4", "Physics", "B", "A*");
```

This creates a Subject object with "PHY4" as the subject ID, "Physics" as the subject name, "B" as the actual grade and "A*" as the expected grade.

Properties

GetActualGrade

This method retrieves the subject's actual grade.

Declaration

```
public string GetActualGrade { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetActualGrade.

Examples

```
Subject sub = new Subject("PHY4", "Physics", "B", "A*");
sub.GetActualGrade;
```

This will return "B".

GetExpectedGrade

This method retrieves the subject's expected grade.

Declaration

```
public string GetExpectedGrade { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetExpectedGrade.

Examples

```
Subject sub = new Subject("PHY4", "Physics", "B", "A*");
sub.GetExpectedGrade;
```

This will return "A*".

GetSubjectID

This method retrieves the subject ID.

Declaration

```
public string GetSubjectID { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetSubjectID.

Examples

```
Subject sub = new Subject("PHY4", "Physics", "B", "A*");
sub.GetSubjectID;
```

This will return "PHY4".

GetSubjectName

This method retrieves the subject name.

Declaration

```
public string GetSubjectName { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetSubjectName.

Examples

```
Subject sub = new Subject("PHY4", "Physics", "B", "A*");
sub.GetSubjectName;
```

This will return "Physics".

Methods

CompareTo(Subject)

This method compares one Subject object with another Subject object.

Declaration

```
public int CompareTo(Subject other)
```

Parameters

Type	Name	Description
Subject	*other*	A different Subject object.

Returns

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

Examples

```
Subject sub = new Subject("MAT3", "Maths", "A", "B");
Subject sub2 = new Subject("MAT3", "Maths", "A", "B");
sub.CompareTo(sub2);
```

This returns `0`, as the objects have the exact same subject ID.

ToString()

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

Declaration

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

Returns

Type	Description
System.String	The string representation of the Subject object.

Overrides

System.Object.ToString()

Examples

```
Subject sub = new Subject("BIO3", "Biology", "C", "B");
sub.ToString();
```

Implements

System.IComparable<T>