Class Student

This class defines a `Student` object.

Implements IComparable to allow Student objects to be compared.

Author: Yashwant Rathor

Inheritance

- System.Object
  - Student

Implements

System.IComparable<Student>

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 Student : IComparable<Student>
```

Constructors

Student(String, String, String, Int32, String)

This custom constructor is responsible for creating a Student object.

Declaration

```
public Student(string ID, string f_name, string l_name, int year, string clName)
```

Parameters

Type	Name	Description
System.String	*ID*	A student's unique 4-digit ID number.
System.String	*f_name*	A student's first name.
System.String	*l_name*	A student's last name.
System.Int32	*year*	A student's current school year.
System.String	*clName*	A student's class name.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
```

This creates a Student object with the ID value as "0007", first name as "John", last name as "Smith", school year as '9' and class name as "Grey".

Properties

GetClassName

This method retrieves the student's current class name.

Declaration

```
public string GetClassName { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetClassName.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetClassName;
```

This will return "Grey".

GetFilePath

This method retrieves the unique filepath to be used to store the student's report.

Declaration

```
public string GetFilePath { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetFilePath.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetFilePath;
```

This will return "C:/Student Results/Year 9/John Smith.txt".

GetForename

This method retrieves the student's first name.

Declaration

```
public string GetForename { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetForename.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetForename;
```

This will return "John".

GetFullName

This method retrieves the student's full name.

Declaration

```
public string GetFullName { get; }
```

Property Value

Type	Description
System.String	Returns the concatenated value of GetForname and GetSurname

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetFullName;
```

This will return "John Smith".

GetStudentID

This method retrieves the student's ID.

Declaration

```
public string GetStudentID { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetStudentID.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetStudentID;
```

This will return "0007".

GetSubjects

This method retrieves all subjects that the student studies.

Declaration

```
public List<Subject> GetSubjects { get; }
```

Property Value

Type	Description
System.Collections.Generic.List<Subject>	Returns the value of GetSubjects.

Examples

```
Student sd = new Student("5451", "Dylan", "Thomas", 10, "Sapphire");
sd.AddSubjectToStudent(new Subject("FRE3", "French", "A", "A"));
sd.AddSubjectToStudent(new Subject("BIO3", "Biology", "C", "B"));
sd.AddSubjectToStudent(new Subject("PSY3", "Psychology", "B", "B"));
sd.GetSubjects;
```

This will return '{("FRE3", "French", "A", "A"), ("BIO3", "Biology", "C", "B"), ("PSY3", "Psychology", "B", "B")}'.

GetSurname

This method retrieves the student's last name.

Declaration

```
public string GetSurname { get; }
```

Property Value

Type	Description
System.String	Returns the value of GetSurname.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetSurname;
```

This will return "Smith".

GetYear

This method retrieves the student's current school year.

Declaration

```
public int GetYear { get; }
```

Property Value

Type	Description
System.Int32	Returns the value of GetYear.

Examples

```
Student sd = new Student("0007", "John", "Smith", 9, "Grey");
sd.GetYear;
```

This will return '9'.

Methods

AddSubjectToStudent(Subject)

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

Declaration

```
public void AddSubjectToStudent(Subject s)
```

Parameters

Type	Name	Description
Subject	*s*	Subject object to be added to the student.

Examples

```
Student sd = new Student("0021", "Jane", "Harper", 12, "Navy");
sd.AddSubjectToStudent(new Subject("PSY4", "Psychology", "C", "A"));
```

A new subject has been added to 'sd'.

CompareTo(Student)

This method compares one Student object with another Student object.

Declaration

```
public int CompareTo(Student other)
```

Parameters

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

Returns

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

Examples

```
Student sd = new Student("2000", "Dean", "Ambrose", 8, "Crimson");
Student sd2 = new Student("2000", "Dean", "Ambrose", 8, "Crimson");
sd.CompareTo(sd2);
```

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

GetSubject(Int32)

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

Declaration

```
public Subject GetSubject(int index)
```

Parameters

Type	Name	Description
System.Int32	*index*	The index position of the desired Subject object.

Returns

Type	Description
Subject	Returns the subject located at the index position specifed by the 'index' parameter.

Examples

```
Student sd = new Student("2000", "Dean", "Ambrose", 8, "Crimson");
sd.AddSubjectToStudent(new Subject("SOC4", "Sociology", "A", "B"));
sd.AddSubjectToStudent(new Subject("GMN4", "German", "B", "B"));
sd.AddSubjectToStudent(new Subject("MAT4", "Maths", "A", "A*"));
sd.GetSubject(0);
```

The subject at index position 0 (Sociology) will be returned.

GetTargetCount(String)

This method retrieves the total number of subjects that the Student exceeds, meets or falls belows their expected grades.

Declaration

```
public int GetTargetCount(string type)
```

Parameters

Type	Name	Description
System.String	*type*	

Returns

Type	Description
System.Int32	The total number of subjects that satisfy the 'type' string parameter are returned.

Examples

```
Student sd = new Student("3650", "Oliver", "Ricardo", 9, "Copper");
sd.AddSubjectToStudent(new Subject("MAT3", "Mathematics", "A*", "B"));
sd.AddSubjectToStudent(new Subject("BIO3", "Biology", "B", "B"));
sd.AddSubjectToStudent(new Subject("CHM3", "Chemistry", "A", "B"));
sd.AddSubjectToStudent(new Subject("PHY3", "Physics", "C", "B"));
sd.AddSubjectToStudent(new Subject("ENG3", "English", "B", "A"));
```

```
sd.GetTargetCount("above");
```

This will return '2'.

```
sd.GetTargetCount("equal");
```

This will return '1'.

```
sd.GetTargetCount("below");
```

This will return '2'.

RemoveDuplicateSubjects()

This method removes any duplicate subjects inside the Student object.

Declaration

```
public void RemoveDuplicateSubjects()
```

Examples

```
Student sd = new Student("5001", "Tom", "Curry", 10, "Purple");
sd.AddSubjectToStudent(new Subject("GEO3", "Geography", "A", "B"));
sd.AddSubjectToStudent(new Subject("HIS3", "History", "B", "A*"));
sd.AddSubjectToStudent(new Subject("GEO3", "Geography", "A", "B"));
sd.RemoveDuplicateSubjects();
```

The second occurence of 'Geography' will be removed as it already exists once.

RemoveSubject(Int32)

This method removes a subject at a particular index position, inside the Student object.

Declaration

```
public void RemoveSubject(int i)
```

Parameters

Type	Name	Description
System.Int32	*i*	The index where the subject is to be removed from.

Examples

```
Student sd = new Student("2608", "Bradley", "Wiley", 13, "Bronze");
sd.AddSubjectToStudent(new Subject("FRE3", "French", "B", "A"));
sd.AddSubjectToStudent(new Subject("CHM3", "Chemistry", "A*", "A"));
sd.AddSubjectToStudent(new Subject("ENG3", "English", "C", "B"));
sd.RemoveSubject(2);
```

The subject at index position 2 (English) will be removed.

ToString()

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

Declaration

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

Returns

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

Overrides

System.Object.ToString()

Examples

```
Student sd = new Student("7142", "Ralph", "Dibney", 11, "Silver");
sd.ToString();
```

Implements

System.IComparable<T>