

Why don't these methods have a return type?

```
public Grade(int studentID, string assessmentDate, string subject, string  
assessment, string comments)  
{  
    StudentID = studentID;  
    AssessmentDate = assessmentDate;  
    SubjectName = subject;  
    Assessment = assessment;  
    Comments = comments;  
}
```

```
// Default constructor
```

```
public Grade()  
{  
    StudentID = 0;  
    AssessmentDate = DateTime.Now.ToString("d");  
    SubjectName = "Math";  
    Assessment = "A";  
    Comments = String.Empty;  
}
```

What's this?

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

How would you explain this?

```
Subjects = new List<string>() { "Math", "English", "History", "Geography", "Science" };
```

How would you explain this? What is it?

```
public string AssessmentDate
{
    get
    {
        return _assessmentDate;
    }

    set
    {
        DateTime assessmentDate;

        // Verify that the user has provided a valid date
        if (DateTime.TryParse(value, out assessmentDate))
        {
            // Check that the date is no later than the current date
            if (assessmentDate > DateTime.Now)
            {
                // Throw an ArgumentOutOfRangeException if the date is after the current date
                throw new ArgumentOutOfRangeException("AssessmentDate", "Assessment date must be on or before the current date");
            }

            // If the date is valid, then save it in the appropriate format
            _assessmentDate = assessmentDate.ToString("d");
        }
        else
        {
            // If the date is not in a valid format then throw an ArgumentException
            throw new ArgumentException("AssessmentDate", "Assessment date is not recognized");
        }
    }
}
```

How would you explain this?

```
[TestInitialize]
```

```
public void Init()
```

```
{
```

```
// Create the data source (needed to populate the Subjects collection)
```

```
GradesPrototype.Data.DataSource.CreateData();
```

```
}
```

```
[TestMethod]
```

```
public void TestValidGrade()
```

```
{
```

```
    GradesPrototype.Data.Grade grade = new GradesPrototype.Data.Grade(1, "01/01/2012", "Math", "A-", "Very good");
```

```
    Assert.AreEqual(grade.AssessmentDate, "01/01/2012");
```

```
    Assert.AreEqual(grade.SubjectName, "Math");
```

```
    Assert.AreEqual(grade.Assessment, "A-");
```

```
}
```

What is this method? Why is it here?

```
public int CompareTo(Student other)
{
    // Concatenate the LastName and FirstName of this student
    string thisStudentsFullName = LastName + FirstName;

    // Concatenate the LastName and FirstName of the "other" student
    string otherStudentsFullName = other.LastName + other.FirstName;

    // Use String.Compare to compare the concatenated names and return the result
    return(String.Compare(thisStudentsFullName, otherStudentsFullName));
}
```