

C++	C#
<pre> namespace system { class date { public: // Constructs to today's date. date(); date(int Year, int Month, int DayOfMonth); bool IsValid() const noexcept; int GetYear() const noexcept; void SetYear(int Year); int GetMonth() const noexcept; void SetMonth(int Month); int GetDayOfMonth() const noexcept; void SetDayOfMonth(int DayOfMonth); private: // Internal details }; } int main() { system::date date(2018, 15, 5); // Oops... system::date date2(2018, 5, 31); assert(date.IsValid()); // Fails assert(date2.IsValid()); // Passes // Create a new date that is ten days later system::date date3(date2.GetYear(), date2.GetMonth(), date2.GetDayOfMonth() + 10); date2.SetMonth(1); return 0; } </pre>	<pre> namespace Time { class Date { // Constructs to today's date. public Date() { // Implementation goes here } public Date(int year, int month, int dayOfMonth) { this.Year = year; this.Month = month; this.DayOfMonth = dayOfMonth; this.IsValid = date.InternalIsValid(year, month, dayOfMonth); } public bool IsValid { get; private set; } public int Year { get; set; } public int Month { get; set; } public int DayOfMonth { get; set; } private static bool InternalIsValid(int year, int month, int dayOfMonth) { // Implementation goes here. return dayOfMonth < 32; } } } public class Program { public static void Main() { var date = new Time.Date(2018, 15, 5); // Oops... var date2 = new Time.Date(2018, 5, 31); System.Diagnostics.Debug.Assert(date.IsValid); // Fails System.Diagnostics.Debug.Assert(date2.IsValid); // Passes // Create a new date that is ten days later var date3 = new Time.Date(date2.Year, date2.Month, date2.DayOfMonth + 10); date2.Month = 1; } } </pre>