Brioso, Ren

All tests are open book.

Read the book.

First assignment due Friday after next, September 4th

Submit assignments as a zip or an exe

Sergeant rb

Assignments are actually due two weeks after listed dates (don’t rely on this bc end of semester crunch)

For next week, play around with downloadables, read ch.1, 2, 3

Chapter 2 notes

Types of controls:

Label- displays text on a form

Text box- lets the user enter text on a form

Button initiates form processing when clicked.

When naming controls, begin with a two to three letter prefix to show the typ of control. Example: txtName or lblTitle

Tab order should be set from left to right, top to bottom. (tab index starts at 0)

Access keys are keyboard shortcuts. Set by entering the intended key into the text property, preceded by an &.

**Chapter Three Notes**

Method of an object-

determine the operations that can be performed by the object

events-

signals sent by the object to your application that can be responded to.

Members-

properties, events, and methods of an object

class- code that defines the characteristics of an object

object- an instance of a class.

To refer to an object within the code, type the name of the object, a dot, and then the member (or property) you’re referencing. <object>.<Text> or <object>.<TabIndex>

Sytax for deploying a method directly from a class onto an object. (static member)

<Class>.<method>(<object>.<member>)

Static Member-

A member that can be referred to

**Q: ARE METHODS JUST MEMBERS FOR A CLASS INSTEAD OF AN OBJECT?**

Windows forms applications are event driven.

Event hander- a special block of code that handles the occurrence of an event.

Event wiring- making an event handler

Common Control Events:

Click: user clicks the control

DoubleClick: user double clicks the control

Enter: focus is moved to the control

Leave: focus is moved from the control to somewhere else

Common form events:

Load: Form is loaded into memory

Closing: the form is in the process of closing

Closed: the form is closed.

Creating event handlers from the form designer:

Double click the object in the form designer

Event Syntax:

<object>\_<Event>

Comments: single line- // this is an example of a small comment.

Big comments:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*This would be an example of a big

\*Freakin’

\*comment

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.NET is a system software that Windows runs in the background.

If you’re in an event handler for an object, you have to use “this” to refer to the object.

Data does not go into an object. It goes into a property of an object.

Compound conditional: && or & or | or ||

.ToString(“p1”)= converts a number to a string with only one decimal place.

Decimals can be implicitly converted. Integers cannot be implicitly converted.

**Before next week’s class: Read chapter 4 and 11, do assignments 2 and 3,**

**9-10-20 Lecture Notes**

There are Two Assignments for next week. (4-1, and 4-2)

**Chapter Four Notes**

There are eight data types used to store integers.

Int works for short integers, but use long if you have a lot of decimal places.

Use short or byte keyword data types for very small ints to save on memory.

Double and float are less accurate than the decimal type, but also use less memory.

Use char to store a single character. Ex: ‘a’

Use bool to store a true or false value.

A screenshot of a cell phone

Description automatically generated

Short hand for assigning literal data types:

Decimal: <varName> = <value>m

Scientific notation: <varName> = <value>e-<\* 10 to the nth power >

Declare and initialize two variables of the same type with a single statement:

int x = 0, y = 1;

const decimal salesTax = 0.75m (“m” for money)

converting to other literal data types:

Casting: widening conversions, performed automatically to make a less precise value more precise.

Explicit cast: you need to actually perform a conversion to narrow the precision of a value.

A value is truncated rather than rounded to narrow precision.

Ex: int grade = (int) 93.75; grade will then equal 93.

Ex:

int b = 86, c = 91;

decimal result = (decimal)b / (decimal)c;

**Importing the math class:**

the round method uses banker’s rounding where everything that ends with the digit 5 is rounded to an even number, so 23.85 would be rounded down to 23.8

A screenshot of text

Description automatically generatedWorking with strings:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

ToString([format]): converts a numeric value to a string with an optional given format that indicates precision.

Parse(): converts a string to a numeric value; raises an exception if there are any non-numeric chars in the string

TryParse(): returns true if Parse() can be run on the given string, otherwise returns false.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Enumerations**

A set of related constants that define a value type where each constant is known as a member of the enumeration.A screenshot of a cell phone

Description automatically generated

To set a numeric data type (value type) to null, type ‘?’ right after the type keyword

Ex: int? counter = null;

HasValue method returns a Boolean value based on whether or not the variable has a null or not.

Syntax for while loop:

while (<condition>)

{

<statements>

}

Syntax for do-whille loops:

do

{

<statements>

}

while (<condition>)

For loops:

Start with the ‘for’ keyword followed by three args enclosed in parentheses, separated by semicolons

Arg1: counter variable declaration and initialization. Data type must be declared.

Arg2: boolean expression to say when the loop will run. Ex: i < 10

Arg3: how the counter variable will change after each iteration.

Syntax:

For (int i = 0; i > 15; i++)

{

<statements>

}

Jump statements can work to skip code within a loop.

Break statements go to the end of the loop.

Continue statement jumps back to the start of a loop.