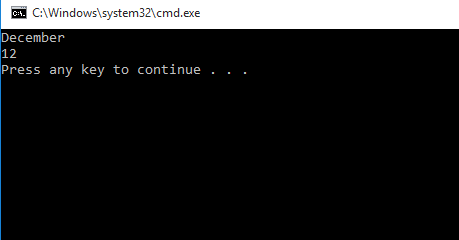
Student: Brian Johnston

Class:COP2362

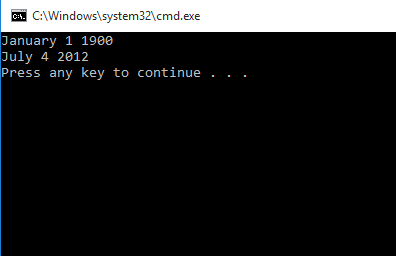
Assignment Tutorial 2-3

I worked alone

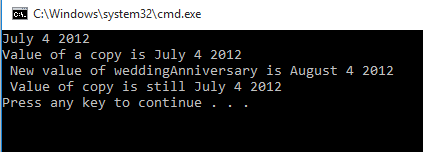
Screen Shot Page 218 Step 13



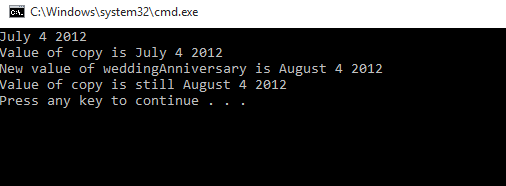
Page227 step 10



Page 229 Step 7



Page 230 Step 11



**Code**

Program.cs

#region Using directives

using System;

using System.Collections.Generic;

using System.Text;

#endregion

namespace StructsAndEnums

{

class Program

{

static void doWork()

{

// Month first = Month.December;

// Console.WriteLine(first);

// first++;

// Console.WriteLine(first);

// Date defaultDate = new Date();

// Console.WriteLine(defaultDate);

Date weddingAnniversary = new Date(2012, Month.July, 4);

Console.WriteLine(weddingAnniversary);

Date weddingAnniversaryCopy = weddingAnniversary;

Console.WriteLine("Value of copy is {0}", weddingAnniversaryCopy);

weddingAnniversary.AdvanceMonth();

Console.WriteLine("New value of weddingAnniversary is {0}", weddingAnniversary);

Console.WriteLine("Value of copy is still {0}", weddingAnniversaryCopy);

}

static void Main()

{

try

{

doWork();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

}

Month.cs

using System;

namespace StructsAndEnums

{

enum Month

{

January, February, March, April,

May, June, July, August,

September, October, November, December

}

}

Date.cs

using System;

namespace StructsAndEnums

{

class Date

{

private int year;

private Month month;

private int day;

public Date(int ccyy, Month mm, int dd)

{

this.year = ccyy - 1900;

this.month = mm;

this.day = dd - 1;

}

public override string ToString()

{

string data = String.Format("{0} {1} {2}", this.month, this.day + 1, this.year + 1900);

return data;

}

public void AdvanceMonth()

{

this.month++;

if (this.month == Month.December + 1)

{

this.month = Month.January;

this.year++;

}

}

}

}