Object-Oriented Programming

Session: Week 6 Session 1
Instructor: Eric Pogue

Agenda:

1. Review this week's programming Assignment
2. Introduce the week's Learning Objectives
3. Topics

FastPrime in C#

Write a performance optimized command line C# application that will programmatically find prime numbers and store those numbers sorted in an output file.

In FastPrime we will create a command line Java application that will:

1. Use multiple threads to find the prime numbers between two numbers
2. Sort those results and store them to a file
3. Perform some timings
4. … And do this all very fast

See the details in this week's assignment

Learning Objectives – Week 6

1. Download and install Visual Studio... And implement Hello World in C#
2. Identify characteristics of Java, Python, and C#
3. Professional positioning
4. ~~Write non-object-oriented programs that use sequence, selection, and repetition~~
5. Define a C# class, complete with properties, methods, and constructors
6. Use C# built-in text file objects to create and read text files
7. Use inheritance to create a hierarchy of classes that are related to each other
8. Create objects of classes and use them to carry out the work of your program
9. Work with C# lists
10. Deal with a list of related objects polymorphically

Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. It can produce both native code and managed code. [link]

- Download and install Visual Studio and C# [link]
- Download the Visual Studio Community 2017 edition with C# selected as your primary environment

Object-Oriented Languages and Tools

The TIOBE index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The full TIOBE is available online [link].

TIOBE Index for March 2017:

Mar 2017	Mar 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.384%	-4.14%
2	2		C	7.742%	-6.86%
3	3		C++	5.184%	-1.54%
4	4		C#	4.409%	+0.14%
5	5		Python	3.919%	-0.34%
6	7	↑	Visual Basic .NET	3.174%	+0.61%
7	6	↓	PHP	3.009%	+0.24%
8	8		JavaScript	2.667%	+0.33%
9	11	↑	Delphi/Object Pascal	2.544%	+0.54%
10	14	⇈	Swift	2.268%	+0.68%

We will utilize mostly Java and C# for our object-oriented programming examples. We may (or may not) do any Python work. Since it is often 'unnatural' to show procedural programming examples in Java, C#, or Python, we will implement programs in C to demonstrate procedure programming examples. Let me know if you have a desire to do some Python work… or work in another OOP language. If so, we can likely work something out.

Note that our reluctance to utilize C++ as a OOP learning tool is does not diminish the value of the C++ toolset. However, C++ is generally considered a very powerful set of tools with a steep learning curve. It's a very sharp knife… use it carefully.

Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. [link]

- Compiles to Java byte codes that run in the Java Virtual Machine (VM)
- Achieved portability by running in VMs that exist on many platforms
- Achieved dominance in the enterprise and for server side development
- Plays a center role in Android development*
- Served as a platform for multiple additional languages have been developed to compile to Java bytecode and run in the Java VM including Groovy and Scala
- Achieved only minimal success in the development of commercial applications or applets
- Syntax Notes: strongly typed, object-oriented, single inheritance, interface focused

*The history of Google's Android and Java is nothing short of a soap opera: http://www.zdnet.com/article/the-real-history-of-java-and-android-as-told-by-google/

Android uses mostly the Java API but does not have the right to use the Java name. Be aware that Software and Legal Protections can be VERY complicated.

Python

An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. [link]

- Achieves portability through interpreter running on various platforms
- Achieved great success as a “quick-and-dirt” scripting tool
- … and in the data sciences realm
- Runs slower because it is interpreted
- Shares a similar space to other scripting languages like Perl and PowerShell
- Syntax Notes: loosely typed, indent sensitive (no brackets or semi-colons), object-oriented

C#... And .NET

C# (pronounced “see sharp”) is a general purpose programming language that implements strong typing, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative. [link]

- Compiles to .NET Common Language Runtime (CLR)
- Portable between CLR implementations… but MS Windows focused
- Portable between Windows desktop, cloud, XBOX, tablets, embedded, and phone
- Focused on industrial strength client applications with a solid server-side presence
- Syntax Notes: strongly typed, object-oriented, single inheritance

C#... And .NET (continued)

.NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library and provides language interoperability (each language can use code written in other languages). Programs written for .NET execute in a software "managed code" environment named Common Language Runtime (CLR). [link]

- Implementations for many languages are available for .NET and CLI including Python… but not "real" Java [link]
- Provides and object-oriented platform that is language agnostic
- ASP.NET is the "standard" .NET Web development environment
- Achieving dominance on Windows desktop, competitive place in cloud (Azure), and XBOX

C#... And .NET (continued)

.NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library and provides language interoperability (each language can use code written in other languages). Programs written for .NET execute in a software "managed code" environment named Common Language Runtime (CLR). [link]

- Implementations for many languages are available for .NET and CLI including Python… but not "real" Java [link]
- Provides and object-oriented platform that is language agnostic
- ASP.NET is the "standard" .NET Web development environment
- Achieving dominance on Windows desktop, competitive place in cloud (Azure), and XBOX

Professional Positioning

Become a professional software developer with depth in on or two areas (i.e. an expert in web application development in Java) and an understanding of several more (i.e. know a little system admin / devops and something about the healthcare industry).

- Do NOT become the Java expert (only)
- Languages and environments are tools, you will need to know several and be able to learn more
- Branch out and play other related roles like business analyst, project manager, team leader, database analyst, product manager, architect, etc.
- Know something about the domain
- Get out and see your customers and business partners

End of Session

Course Number: CPSC-24500

Week: 6

Session: 1

Instructor: Eric Pogue

Object-Oriented Programming

Session: Week 6 Session 2
Instructor: Eric Pogue

LEWIS UNIVERSITY
A Catholic and Lasallian University

1. Download and install Visual Studio... And implement Hello World in C#
2. Identify characteristics of Java, Python, and C#
3. Professional positioning
4. ~~Write non-object-oriented programs that use sequence, selection, and repetition~~
5. Use Visual Studio 2017 to define C# classes, complete with properties, methods, and constructors… and much more
6. Create classes and objects to carry out the work of your program
7. Use inheritance to create a hierarchy of classes that are related to each other
8. Use .Net (C#) built-in text file objects to create and read text files
9. Work with .Net (C#) lists
10. Deal with a list of related objects polymorphically

But First… FastPrime C#

Write a command line C# application in Visual Studio 2017 that will programmatically find prime numbers and store the numbers sorted in an output file.

In FastPrime we will create a command line C# application that will:

1. Utilize C#
2. Utilize Visual Studio and the integrated environment & debugger
3. Use multiple threads to find the prime numbers between two numbers
4. Sort those results and store them to a file
5. Perform some timings
6. Come to our Thursday lunch session with any questions… or email your questions head of time

See the details in this week's assignment.

This time we are primarily focused on using a familiar topic (performance, threading, and prime numbers) to learn a new environment and language.

Example: Visual Studio, C#, and Random Numbers

ThreadedRandomNumbers: Calculate 1,000,000,000 random numbers between 1 and 2,000,000. Print “We found number 1024!” to the console each time 1024 is generated. We would expect it to come up approximately 500 times.

- Write a single threaded application
- Divide the application into multiple threads and repeat
- Consider the diminishing returns of adding additional threads

Progressing through:

HelloWorldCS

RandomNumbersCS

RandomNumbersThreadedCS

Once again, this time we are primarily focused on using a familiar topic (performance, threading, and prime numbers) to learn the Visual Studio environment and C# language.

Visual Studio & C#

Let's make sure that we understand:

- Visual Studio 2017… embrace your IDE
- Visual Studio New Application Wizard*… mixed blessings
- Visual Studio Text Editor
- Debugger!
- Debug vs. Release builds
- Project and source code files
- C# Hello World
- Command line arguments in debugger
- Debug and Release configurations
- Text formatting options ({)
- Full screen text editing (shift+alt+enter)

If you are going to adopt an IDE, understand how it works and embrace it's preferences whenever possible.

New Application Wizard's are great, but not nearly as great as they first appear. You must invest the time to understand their "magic" so that when there is a problem you know where to look.

*Important: For this week's assignment you will want to start with File->New->Project->Visual C#->Console App (.NET Framework)… And name your project "FastPrimeCS"

Command Line Arguments: Right click "ConsoleApp1"and select Properties->Debug->Command line arguments.

Bracket Formatting: Tools->Options->Text Editor->C#->Code Style->Formatting->New Lines->Place open

brace on new line for types (unmark all)

Hello World C#:

https://msdn.microsoft.com/en-us/library/aa288463(v=vs.71).aspx

Visual Studio & C# (continued)

- System.Console.WriteLine
- Classes
- Methods
- Properties, Setters, and Getters
- Framework classes.. .NET classes and not C# classes*
- Time & Duration
- Threads

Many Visual Studio languages other that C# (which grew up with .NET) have access to multiple (often competing) class libraries. This can be a substantial challenge as the implementations of similar classes are often not compatible with each other or other elements of their respective libraries. For example, there is at least four competing implementations of "string" that exists in C++ (C, C++, Windows API, MFC, .NET, etc.).

In many cases knowing a library or set of APIs has become more important than a given language.

C# vs. Java GetRandomNumbers Class

```java
// Java
class GetRandomNumbers {
    private Random myRandom;

    GetRandomNumbers() {
        myRandom = new Random();
    }

    public int GetNumberBetween(int min, int max) {
        return min + myRandom.nextInt(max-min);
    }

    public void Find1024() {
        long timesToLookFor1024 = 1000000000;
        long timesFound = 0;

        for (long i=0; i<timesToLookFor1024; i++) {
            int random = GetNumberBetween(1,2000000);
            if (random==1024) {
                timesFound++;
                System.out.format("RandomNumber:%,d Found:%d\n",random, timesFound);
            }
        }
    }
}
```

```csharp
// C#
class GetRandomNumbers {
    Random myRandom;

    public GetRandomNumbers() {
        myRandom = new Random();
    }

    public int GetNumberBetween(int min, int max) {
        return min + myRandom.Next(min, max);
    }

    public void Find1024() {
        long timesToLookFor1024 = 1000000000;
        long timesFound = 0;

        for (long i = 0; i < timesToLookFor1024; i++) {
            int random = GetNumberBetween(1, 2000000);
            if (random == 1024) {
                timesFound++;
                Console.WriteLine("RandomNumber:{0} Found:{1}", random, timesFound);
            }
        }
    }
}
```

Shockingly similar code.

Object-Oriented Programming

Session: Week 6 Session 2 Plus
Instructor: Eric Pogue

1. Download and install Visual Studio... And implement Hello World in C#
2. Identify characteristics of Java, Python, and C#
3. Professional positioning
4. ~~Write non-object-oriented programs that use sequence, selection, and repetition~~
5. Use Visual Studio 2017 to define C# classes, complete with properties, methods, and constructors… and much more
6. Create classes and objects to carry out the work of your program
7. Use inheritance to create a hierarchy of classes that are related to each other
8. Use C# built-in text file objects to create and read text files
9. Work with C# lists
10. Deal with a list of related objects polymorphically

Review: Serialization and Writing/Reading Text Files (IO)

When we were working with Java we learned that… Serialization is an object-oriented programming term that means converting an object to a byte steam usually to be written to or read from a text or binary file.

To write to a Java text file:

- Create a File object, feeding the file's path to the File class constructor
- Create a FileWriter to access the File
- Create a BufferedWriter to write data to the FileWriter efficiently
- Use BufferedWriter's write and newLine functions to commit the data to the file.

To read from a Java text file:

- Create a File object, feeding the file's path to the File class constructor
- Attach a Scanner object to it
- Use Scanner's readLine and hasNextLine functions to read the file

A buffered writer or buffered stream efficiently organization reads and writes for optimal disk performance. The danger is that if there is a power failure (or someone accidentally kicks your power strip) you could lose data. It is good to "flush the buffer" at critical times when using a buffered writer on mission critical projects. There is little or no danger in using a buffered reader.

Binary files versus text files.

XML – Text file format for structured data… HTML for data.
JSON – "Simplified" text file format for structured data.
Binary

.NET (C#) Writing & Reading Files

There a multiple C# classes and (static) methods that we can use to write and read files. We will be looking at several including:

- System.IO.File.WriteAllLines
- FileStream: Used to read from and write to any location in a file
- File Class [link]… get used to looking these things up online

.NET (C#) Lists

The .NET framework offers multiple implementations of Collections and Collections.Concurrent including:

- List [link]: Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.
- ArrayList [link]: Implements the IList interface using an array whose size is dynamically increased as required.
- ConcurrentQueue [link]: Represents a thread-safe first in-first out (FIFO) collection.

Become comfortable in looking at the online documentation and examples… be aware of “reputable” sources and conflicting documentation.

.NET ArrayList Thread Safety: Any public static (Shared in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

If you run into threading issues with ArrayaList in your assignment, you will need to switch over to a “Concerrent” collection list “ConcurrentQueue”.

End of Session

Course Number: CPSC-24500

Week: 6

Session: 2

Instructor: Eric Pogue

Object-Oriented Programming

Session: Week 6 Session 3 (Lunch & Learn)
Instructor: Eric Pogue

LEWIS UNIVERSITY
A Catholic and Lasallian University

Agenda:

1. Review FastPrime C# Assignment
2. FastPrimeLite

FastPrimeLite: I don't want this week's assignment to be difficult because of finding prime numbers and threading! This is intended to be an exercise in taking some things that we know like finding prime numbers and threading and becoming comfortable with implementing them in Visual Studio and C#. As such, my FastPrimeLite application that I am going to show today is going to be very much like what you will need to do from a logic perspective for this week's assignment. You should feel free to copy (well type it in yourself, update the variable, and get the syntax correct) the code for you to become familiar with Visual Studio and C# editing, debugging, and updating.

FastPrime C#

6. FastPrime C# Assignment (20 points). This should look familiar. It is the same application we wrote last week ported to C#.

Write a command line C# application that will programmatically find prime numbers [link] and store those numbers sorted in an output file.

#	Requirement	Points
1	The application must be developed entirely in C#, compile under Visual Studio 2017, and run in a standard windows command line environment.	6
2	Take in two command line arguments that represent the start and end number to consider. The application should find all prime numbers within the inclusive range. For example, I could pass in 2 and 1,000,000. In addition, the application should fail gracefully with a meaningful error message if inappropriate arguments are passed into the application.	3
3	The application should utilize multiple threads.	2
4	An output file should be created in the current folder and named FastPrime.txt or FastPrime.bin (if you chose to store the prime numbers as binary). I should contain a sorted list of the prime numbers that were found. If you use a text file, it should contain one number per line with no other characters. Binary files should include ONLY a sorted list of integers.	3

FastPrime C# (continued)

5	Each prime number should be printed to the console window when it is found along with a reference to the thread that found the number. After the program has completed, it should print to the command prompt (1) the number of prime number found (and stored in the file), (2) the start time, (3) the finish time, and (4) elapsed time. All should be valid, reported to the second, and displayed in a visually appealing format.	2
6	The application should be FAST, this will be scored relative to other timings.	2
7	Submit two files. The first file should be the single C# file named "Program.cs" that was used in the application. You should include your full name in a comment at the beginning of the C# file that you submit. The second file should be the release executable called "FastPrime.exe". You should have testing this executable from the command line and verified that it worked before submitting it to me.	2

FastPrimeLite

Features:

1. Develop application entirely in Visual Studio 2017 and C#
2. Take in two command line arguments that represent the start and end number
3. ~~Fail gracefully with a meaningful error message if inappropriate arguments are passed in~~
4. Find the prime numbers in the range
5. Add numbers to a list
6. Sort the list
7. Use multiple threads… I will use 3 and you should use at least 4
8. Each prime number should be printed to the console window
9. Write to the console (1) the number of prime number found (and stored in the file), ~~(2) the start time, (3) the finish time,~~ and (4) elapsed time
10. Output results to text file called FastPrimeLite.txt
11. Review the location of Program.cs and the release executable named "FastPrimeLite.exe

I want you to be focused on learning the C# environment and syntax and not on reimplenting FastPrime to be better.
Your application should be your own; however, you're welcome and encouraged to copy significant elements of FastPrimeLite in your assignment. You should type it in yourself, change the names of the classes and variables at a minimum, implement the missing features, use at least four threads, and generally enhance the implementation.
The crossed out items are items that will not be implemented if FastPrimeLite but that you will need to implement separately.

Please don't forget the easy things like your name at the top and start/end times, etc

End of Session

Course Number: CPSC-24500

Week: 5

Session: 3

Instructor: Eric Pogue

End of Session

Course Number: CPSC-24500

Week: 5

Session: 4

Instructor: Eric Pogue