**CPSC-24500: Object-Oriented Programming**

**Homework Week 6**

|  |  |  |
| --- | --- | --- |
| **Name:** |  | **Score \_\_\_\_\_ / 30** |

Instructions: Fill in your name above and please answer the questions below. Submit your answers via the Blackboard homework submission link by the end of the day Sunday, March 26. This is an individual assignment and answers are not to be shared.

You will need to save a copy of the MS Word file to your local drive, fill out the name and date information, and answer each question by highlighting the best answer, by writing your answer after the question, or my inserting an image. Please let me know if you have difficulties.

1. Which of the following is not an advantage of an interpreted language?
   1. It is easier to debug.
   2. It runs more quickly.
   3. It enables you to try out code before you add it to the source code.
   4. It is easier to make cross-platform than a compiled language is.
2. Which of the following is a false statement?
   1. Python is an interpreted language.
   2. C# is most prevalent in the MS Windows environments.
   3. Both Java and C# compile to Java bytecodes
   4. C# is well suited for applications that need to run quickly.
3. Python is a loosely typed language. What does that mean?
   1. You don’t explicitly declare the type of the variables you use in your program.
   2. There are no data types in the Python language.
   3. You can perform any arithmetic operation you want with any variable.
   4. You never need to type-cast variables, as the Python interpreter is able to determine the required data type automatically.
4. Please right three or more paragraphs comparing the similarities and differences between the Java, .Net, and Python runtime environments. I’m looking for about two-thirds of a page describing similarities and differences in the environments and particularly in the way they handle portability and prioritize performance. (5 pts.)

|  |
| --- |
|  |

1. Briefly compare your FastPrime Java and FastPrime C# applications. Include your objective timings and subjective opinion of the performance of the applications. (2 pts.)

|  |
| --- |
|  |

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

Write a performance optimized command line C# application that will programmatically find prime numbers [[link]](https://en.wikipedia.org/wiki/Prime_number) 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 |
| 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 “FastPrime.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 |

If your solution does not compile and execute without errors when it is submitted, you will lose 6 points AND I will send it back to you to fix and resubmit before I attempt to continue grading the assignment.

Do not copy another student’s work. I will use MOSS to detect plagiarism and will not ask for clarification if MOSS concludes you have copied another student’s work.

Tackle this problem gradually and make sure that you review the examples that we cover in class. The main goal of our discussions, lectures, and examples this week are intended to allow you to successfully deliver this application. Also, don’t hesitate to post something on our discussion board or to reach out to me directly if you need assistance.

Pace yourself. Do not attempt to do this in one night.

Good luck!