Week 1 Announcement

Posted Feb 26, 2024 10:52 PM

Hello Students:

This is **Suhash** and I am your **Instructor** for this course **CS 230: Operating Platforms**. In this course, we will develop a deeper understanding of operating platforms and architectures through the analysis and evaluation of the characteristics, advantages, and weaknesses of each. We will also learn the value of utilizing **software design patterns** as well as how to utilize them to solve problems.

This course covers the following competencies, which represent the knowledge and skills relevant to your field:

- Utilize software design templates and patterns to efficiently solve a problem
- Analyze the characteristics of and techniques specific to various systems architectures
- Evaluate the characteristics, advantages, and weaknesses of various platforms

As you read about the components of operating systems, I encourage you to think beyond the platform you are using to develop software. I would also remind you of the distinction between the server (where the application is going to operate) and the platform that a developer uses to develop the application.

I would encourage you to pay particular attention to making connections between operating platforms and servers, as this will be a key concept in future assignments. You will read about this in **Module One** to gain a foundational understanding of the components of operating platforms.

The assigned textbook for this course, ***Hands-On Design Patterns with Java***, illustrates various software design patterns such as ***creational*** *(how objects are created/instantiated, e.g. factory, singleton, etc.)*, ***structural*** *(how objects and classes are combined to form a system, e.g. composite, façade, adapter, etc.)* and ***behavioral*** *(how objects and classes interact with each other in a system, e.g. iterator, observer, strategy, etc.)* patterns (Gang of Four design patterns). In addition to the textbook, there are plenty of materials available online about design patterns, here is one. I would encourage you to search through **YouTube** to find others as needed.

In this course, we will also learn **Unified Modeling Language (UML)** diagramming technique to model a software application. **UML** is intended to provide a standard way of visualizing a design. As the old saying goes, a picture is worth a thousand words. As such, UML diagrams are visual aids that would help Programmers, Software Engineers and Managers understand a program quickly without having to go through possibly thousands of lines of code.

Throughout my past more than 20 years of software development career, I have had the opportunity to leverage some of these established design patterns such as **Singleton, Dependency Injection, Repository, MVVM, MVC, Observer** patterns, among others, and we may discuss them as appropriate, throughout this course.

In addition to Software Design Patterns, SOLID Design Principles are another software engineering concepts that I would encourage you to explore as well.

There will be a quiz in this module.

Please feel free to reach out to me (s.sarkar1@snhu.edu) if you have any questions or concerns.

Thanks,

Suhash