SWE 645
Component-based Software Development

Summer Semester, 2023
Location: Online\Blackboard Collaborate Ultra
Time: Thurs 5:00-8:30pm

Professor:	**Dr. Vinod Dubey**
Email:	vdubey@gmu.edu
Class Hours:	Thurs 5:00-8:30 PM
Prerequisite:	SWE 619, or CS 540 and CS 571, or permission of instructor.
Office Hours:	Anytime electronically, or by an appointment
GTA:	**Arnab Debnath**, Email: adebnath@gmu.edu
GTA Office Hours:	TBD -Please check GTA office hours site https://cs.gmu.edu/academics/gta-office-hours/

Overview

OBJECTIVE:

This class will be a detailed study of the concepts and engineering principles of software component and component-based software systems. After the course, students should be prepared to create large-scale component-based containerized Web applications on AWS platform. The course is designed to provide a hands-on experience in DevOps engineering and core AWS services.

CONTENT:

SWE 645 covers some of the topics related to the software development models that are used to support component-based software systems. We will be studying the software design and development side of component-based software that includes design principles to architect resilient, highly available, performance oriented, and secure applications in AWS cloud. The course is designed to provide a rich full stack development comprising the development

of both front end (client side) and back end(server side) portions of enterprise web applications.

SWE 645 covers some of the topics related to the software development models that are used to support component-based software systems. We will be studying the software design and development side of component-based software that includes design principles to architect resilient, highly available, scalable, and secure applications in AWS cloud. **Students will have options to experiment with Google Cloud Platform (GCP) as well. The course content will largely focus on server-side software design and development. We will learn technologies such as Microservices, Docker Containers, Container Orchestrator - Kubernetes, DevOps – CI/CD, Infrastructure as Code – Terraform, Configuration Management Using Ansible, GitOps, Resilient Architectures in AWS cloud, as well as a range of AWS services ranging from compute, networking, storage, and security. Students will have options to do homework assignments/project on Amazon Web Services (AWS) and/or Google Cloud Platform (GCP).**

Though SWE 619 is the only required prerequisite, other topics such as web based Java programming (HTML, CSS, JavaScript, Servlets, JSPs) and database programming (JDBC) as a background knowledge will be helpful. SWE 642, though not a required prerequisite, is a great background course for these topics. The class will be very practical (how to build things) and require extensive programming assignments.

Textbook and Readings

- Java: How to Program, 9th or 11th edition, Deitel and Deitel , Prentice Hall. (Recommended)
- GitOps and Kubernetes: Continuous Deployment with Argo CD, Jenkins X, and Flux by Billy Yuen, Alexander Matyushentsev, Todd Ekenstam, Jesse Suen (Recommended)
- Kubernetes in Action 1st/2nd Edition by Marko Luksa (Recommended)
- Building Event-Driven Microservices: Leveraging Organizational Data at Scale 1st Edition by Adam Bellemare (Recommended)
- Please refer to references in the reading section of individual topics.

In addition to above, we will read from various sources on the web, and slides that will be made available during the class.

Grading

EXAMS:

There will be a midterm and a final exam, both in class. The final exam will focus on material covered after the midterm.

Phone Use Policy: Phones should be switched off during the mid-term and final exams. No phone of any kind, especially smart phones with Internet access and camera is allowed to be on person during exams.

HOMEWORK: Programming Assignments/Group Projects:

A number of homework assignments will be given. I will discuss each in class and make the assignment available on the Blackboard web site. You will submit your solutions by placing links to the executables on your class web sites and submitting the source files through blackboard. Your solution should contain a zip file containing all of your code and a readme file detailing any installation procedures, or dependencies. I reserve the right to deduct points if the readme and/or zip file is not included with the homework. Please read ALL of the assignment requirements, as they will contain important information, such as procedures, naming conventions, and how the assignment should be deployed. Points can, and will, be deducted from the assignments if it doesn't work, or if the source code is not supplied. Please review your submission before you submit it. Be sure that you are able to access Blackboard, as refinements and hints for the assignments will be posted there.

Changing an assignment after the due date without prior permission will be treated as a late submission. Late submissions carry an automatic 10 percent deduction in grade for each week that it is late. Late assignments will not be accepted after 2 weeks. No homework will be accepted after the start of the last lecture of the semester. Assignments will be graded on the correctness of the code and the adherence to the requirements. A more detailed description on the homework grading will be included with the homework requirements when assigned. I also expect to be able to deploy your assignment on my local laptop. We will use combination of AWS cloud computing platform, Tomcat, Docker Containers, Kubernetes to deploy your homework(s) in Amazon Web Services cloud.

MAKEUPS and LATE ASSIGNMENTS:

Unless arrangements are worked out in advance, missed tests **cannot** be made up, and 10 percent per class meeting will be deducted for late homework

submissions. Under no circumstances will any assignment be accepted after the official end of classes (the start of finals week).

GRADING:

Grades will be (approximately): 32% the midterm, 33% the final, 30% the programming assignments, and 5% class participation/quizzes.

Final averages are assigned a letter grade according to the following ranges:

Percentage	Grade
[99,100]	A+
[92, 98]	A
[90, 91]	A-
[88, 89]	B+
[82, 87]	B
[80, 81]	B-
[78, 79]	C+
[72, 77]	C
[70, 71]	C-
[60, 69]	D
[0, 59]	F

Schedule (subject to change; check regularly)

Week	Date	Lecture topic	Readings	Announcements
1	5/25	**Course overview** Introduction to Component-based Software Development **Computing Platform** Introduction to Amazon Web Services (AWS) cloud Demo using Amazon EC2 and Amazon S3, and Tomcat	• https://www.youtube.com/watch?v=98ya1LiEU00&t=83s&spfreload=5 • https://aws.amazon.com/documentation/ec2/ • https://aws.amazon.com/documentation/s3/	HW1
2	6/01	**DevOps Practices in Software Development**	• Class notes • https://www.docker.com/	

		Microservices, DevOps, CI/CD, Docker Containers, GitOps. Argo CD		
3	6/08	**Container Orchestration** Kubernetes, Rancher	• https://kubernetes.io/ • https://rancher.com/docs/	HW2
4	6/15	**Container Orchestration (Contd)** Kubernetes, Rancher **Deploying on Kubernetes** Helm Charts	• Class notes, • https://kubernetes.io/ • https://rancher.com/docs/	
5	6/22	**Microservices Implementation** SpringBoot and Thymeleaf **Midterm Exam Review**	• Class notes •	
6	6/29	**Midterm Exam (7:20 pm - 10:00 pm)**	• In a classroom/online	
7	7/06	**Designing Resilient Architectures in AWS**	• Class Notes	HW3
8	7/13	**Computing on AWS** AMI, Instance Types, Network setting, Storage configuration, Security Group, Key Pair	Class Note https://www.python.org/ • https://www.w3schools.com • https://aws.amazon.com/sql • https://aws.amazon.com/sns/	
9	7/20	**Networking in the Cloud** VPC, Subnets, IP Addressing, Gateways, Endpoints, VPC Peering, Transit Gateway	• https://activemq.apache.org/ • https://kafka.apache.org/	
10	7/27	**Scalable Deployments in AWS** Elastic Load Balancer, Auto Scaling, DNS/Route 53	• Class notes, https://aws.amazon.com/documentation/dynamodb/	
11	8/03	**Storage and Database Services**	• https://aws.amazon.com/lambda/	

		S3, EBS Volume, EFS, Amazon RDS, DynamoDB **Infrastructure as Code (IAC)** – Terraform; Provisioning cloud services, such as EC2, VPC, subnets using terraform. Final Exam Review	• https://aws.amazon.com/documentation/redshift/	
13	8/10	**Final Exam** **(7:20 pm - 10:00 pm)**	In the class room/online	

Academic Integrity

George Mason's policy concerning student conduct applies. Although students are encouraged to discuss the topics covered in class, all homework assignments, exams, and projects are to be completed individually, unless joint work is explicitly authorized by the instructor. If joint work is authorized, all contributing students must be listed on the submission. Any deviation from this is considered an Honor Code violation, and, as a minimum, will result in failure of the submission and, as a maximum, failure of the class.

Please note that there are two honor code policies: an abstract GMU policy and a more specific departmental policy with regard to code plagiarism, test-taking, etc. The students can find them here: Honor Code Policies

---

Disabilities

If you are a student with a disability and you need academic accommodations, please see me and contact the Disability Resource Center (DRC) at 993-2474. All academic accommodations must be arranged through the DRC.