CS520

Java introduction course. No modules were saved but can find some extra files in the folder.

CS 526

CS 601

CS602

CS622

Module 1- Java object oriented programming techniques(class(entity), inheritance(is a),polymorphism(taking many forms), downcasting (cast), abstract class(can’t be instantiated, subclasses must implement the unimplemented methods declared in abstract class), interface

Module 2- exception and text input handling, try catch block, checked(must be caught or declared) vs unchecked exception, regex, scanner class, Stringbuilder

Module 3- Generics(make code reuse easier and enable detection of errors at compile time rather than at run time), bounded generics, wildcard generics, Collection (

The "Java Collections Framework" includes two types of containers:

Collection objects (e.g., lists, stacks, queues, sets), created from the **Collection** interface,

Map objects (for storing key-value pairs), created from the **Map** interface

), ArrayList LinkedList Vector Stack Queue Set Map.

Module 4- Binary I/O(FileInputStream and FileOutputStream), JavaFX(gui), Lambdas and Streams

Module 5- Concurrency (Describe how concurrency, parallelism and multi-threading work in Java programming.

2.

Create and use threads.

3.

Evaluate thread safety and race conditions.

4.

Use synchronized statements and synchronized methods.

5.

Synchronize threads using locks.

6.

Detect deadlock in programming.

7.

Use Thread pool, Executors, ExecutorService.

Module 6- Database connection in java(

Describe relational database concepts.

Use SQL to create and drop tables and also to retrieve and modify data.

Use the JDBC API to access databases and process *ResultSet*s.

Use *PreparedStatement*s to execute pre-compiled SQL statements.

Use *CallableStatement*s to execute stored SQL procedures.

Use the DatabaseMetaData and ResultSetMetaData interfaces.

Apply the concepts of TCP, IP, and Internet address.

Create servers using server sockets and clients using client sockets.

Implement Java networking programs using stream sockets.

Develop servers for multiple clients.

Send and receive objects on a network.

Project: Personal health information app. Use java oop concepts along with javafx gui and db to create an app that takes user login/register and then their input, show stored results, etc.

CS 669

Module 1 – what is db, fields are cloums and record is a row, data modeling, business rules, sql, table(container to hold data

CREATE TABLE Food\_purchase ( food\_item VARCHAR(64), purchase\_date DATE, price DECIMAL(4,2));), insert data (

INSERT INTO Food\_purchase (food\_item, purchase\_date, price)VALUES ('Sandwich', CAST('25-Aug-2011' AS DATE), 4.15);),

Viewing data (

SELECT food\_item, purchase\_date, priceFROM Food\_purchase;),),

remove, update data, select command, delete table.

Module 2 – RDMS, relationships in rdms, primary, foreign keys, entity relationship model, symbols, conceptual diagram, sql join, unique, many example of db design

Module 3 – Advanced data modeling, rules, specialization, generalization, uml for erd(class and their relations), db normalization, aggregate functions, alter table

Module 4 – Database Design(like sdlc processes) concurrency, locking, recovery

Module 5 – Database Performance Tuning and Query Optimization, nested queries

Module 6 –

Labs: Look at the labs folder to learn more db knowledge

Project: Inventory management system of a business.

CS673

Module 1- Intro to Software Engineering, People Project stakeholders (who are they?),Product [Software, code, documents etc (what are they?)], Project Activities (What are basic common activities?), Process Framework to carry out the activities (how to perform SE activities(agile,waterfall,rad)?), Software Configuration Management(CICD, container, IaC), risk management  
  
Module 2 – Req analysis(func & non func, scope creep), user stories, ECB, basic UML diagrams

Module 3 – Design goals(reusability, flexibility, sufficiency, security, etc), architecture(MVC, tiered, c2p, soA, REST, Microservices, security design, class relationships, cohesion-coupling)  
  
Module 4 – Refactoring and testing TDD (red-green-blue) , unit, integration, system, acceptance, func, non-func, regression testing, api testing, automatic vs manual testing, testing tools(Junit, pytest, etc), Code smells([**https://sourcemaking.com/refactoring/smells**](https://sourcemaking.com/refactoring/smells))

Module 5 – All types of UML diagrams, class relationships, white box testing

Module 6 – Security, Secure Software Development Process Model, OSWAP SAMM, SAST vs DAST, code review, security requirements

* Look into CS673\_review document more information

Project: Movie review web app(full stack deployment in aws with devops)

* Use of python flask for backend code and vanilla js for front end(with html and css)
* Use of pytest to test various component
* Use of aes256 bit encryption for user credential
* Test code by component
* Database connection using python
* Use of github actions to setup CI(implement sast tool, dockerization, terraform to deploy in aws)

CS682

Module 1 – About mission statement, selecting system, functional requirement, nonfunctional requirement, use case, activity diagram

Module 2 – About SDLC process(Waterfall, RAD, agile), risk management

Module 3 – Gathering requirements techniques, user story, state transition diagram, gui

Module 4 – UML diagrams, Entity class, class diagram, sequence diagram, inheritance, aggregation, composition, dependencies, multiplicities

Module 5 – Goals of system design, cohesion(high), coupling(low), packages, DFD, microservices, pseudocode