

Assignment Brief

Module code and title	5CS022 Distributed and Cloud Systems Programming
Module leader	Deepson Shrestha
Assignment name	Portfolio
Assignment type	Portfolio
Assignment weighting	50%
Assignment size	
Submission date	Week 12
Submission method	Canvas
Assignment requirements	
Assessment criteria	<p>LO1 Adapt serial algorithms to operate in a distributed way.</p> <p>LO2 Design, implement and test programs that operate in a distributed way.</p> <p>LO3 Modify programs to run on and deploy on cloud infrastructure.</p> <p>LO4 Evaluate cloud-based solutions with respect to performance, security and other pertinent factors.</p>
Additional instructions	Submit your entire portfolio as a Zip file to Canvas. Do not submit the individual files separately
Professional Body requirements	
University Regulations	<p>University's Academic Regulations: https://www.wlv.ac.uk/about-us/governance/legal-information/policies-and-regulations/academic-regulations/</p> <p>Academic Integrity Policy: https://www.wlv.ac.uk/media/departments/office-of-the-vice-chancellor/documents/Academic-Integrity-Policy-from-2019-20.pdf</p> <p>Performance descriptors: https://www.wlv.ac.uk/media/departments/office-of-the-vice-chancellor/documents/University-Performance-Descriptors.pdf</p>
Support	<p>Student Support and Wellbeing: https://www.wlv.ac.uk/current-students/student-support/student-support-and-wellbeing-ssw/</p> <p>Study Guides: https://www.wlv.ac.uk/lib/skills-for-learning/study-guides/</p> <p>Student Handbook: https://www.wlv.ac.uk/current-students/student-handbook/</p>

	Assessment information: http://www.wlv.ac.uk/assessment You should also refer to your Course and Module Guides
Date by which feedback will be provided	6 weeks after close of submission
Method by which feedback will be provided	Via Canvas.

Assessment

This assessment is a Portfolio for 5CS022 Distributed and Cloud Systems Programming, which accounts for 50% of the module marks.

Deadline

A zip archive of all of your work (source code, binary executable, supporting documents) must be submitted, as one zip file, via Canvas on or before **Week 12**.

Details

Coursework

The coursework will consist of a number of questions that you will have to answer by writing a short research-based report and a number of tasks which you will have to carry out by creating a number of specified programs.

Assignment Brief

1. Using the Akka Actor framework, create a simulation of a bank account with multiple concurrent deposits and withdrawals, over a specified number of transactions.

You should do the following:

- Create a BankAccount class as the Actor to hold the account balance, and to respond to deposit and withdrawal messages.
- Create a Deposit class to serve as the message sent to the BankAccount class to deposit money into the account.
- Create a Withdrawal class to serve as the message sent to the BankAccount class to withdraw money from the account.
- Create a Main class to serve as the simulation running program.
- When the Main class starts running, it should create an Actor of the BankAccount class.
- The BankAccount actor should initialize its balance to £100 on startup and print this balance to the console output.
- The Main program should then create 10 random values between -1000 to 1000.

- If a value is greater than zero, the Main program should create a Deposit message with that amount, and send it to the BankAccount actor, which would then add it to its current balance, and then print out the new balance on the console output.
 - If a value is less than zero, the Main program should create a Withdrawal message with that amount, and send it to the BankAccount actor, which would then subtract it from its current balance, and then print out the new balance on the console output.
 - After all 10 transactions have been processed, the program should terminate.
2. Hosting a web application on a Aws or Azure Linux server. The application should include both a front-end and a back-end. Your task is to create the hosting environment on the server and upload all necessary files for the application to function properly.
 3. In a similar approach to Workshop on Google Firestore, create a “movies” review HTML/JavaScript web app, based on a Google Firestore NoSQL database. The web app should access the movie review data store in a Google Firestore NoSQL database via JavaScript and present the information via HTML on the web browser. The web app should have the ability to add the movie name, a rating score from 0 to 5 (integers only), the director’s name and the release date. It should have the ability to be sorted in order on any of the fields. It should have the ability to edit and delete individual reviews. There is no requirement for any user authentication for the web app. You should host your application on Aws S3 bucket.