INFORMATION TECHNOLOGY

SFDV3006: Concurrent Programming

1. **3 credit hours, 15 weeks, 4 hours per week**

2. **Prescription:** This course examines a range of techniques for programming multi-threaded and distributed applications. Topics include synchronization mechanisms used for programs that communicate via shared memory, and message passing techniques for programs that communicate across a network. Practical work involves implementing programs using these techniques in a modern concurrent language, such as Java.

Concurrent programming paradigms are well established in the areas of operating systems, computer networks and databases. Today they are increasingly found in sophisticated multithreaded or distributed applications, and the ability to construct concurrent programs is becoming an important skill for the modern programmer. Concurrency is also a very rich area of both practical and theoretical study in Computer Science. SFDV3006 will present both the theory of concurrent programming and a range of useful paradigms for the construction of concurrent algorithms

3. **Learning aims**

- To give students an understanding of the benefits and challenges of concurrent programming
- SFDV3006 is designed to help students learn and develop an understanding of concurrent programming and its use in modern computer systems.
- Acquire the knowledge and skill for implementing concurrent execution, shared variables and objects, synchronization primitives, higher level language constructs, verification of concurrent using programs using state models, message passing, remote invocation and a range of well known paradigms for the design of concurrent programs.

Learning Outcomes

On successful completion of SFDV3006 students will be able to:

1. An understanding of the use of key mechanisms for correct concurrent execution using shared variable by using synchronisation primitives and higher level language constructs.

2. An understanding of the key mechanisms for writing correct concurrent programs by implementing and testing their own solution to a problem in Java.

3. Understand the problems of concurrent programs and how to solve them

4. An understanding of mechanisms for building distributed applications.

5. An understanding of the implementation of some key paradigms for distributed and parallel programming (master-worker, tuple spaces) using message passing mechanisms

4. **Graduate attributes addressed**:

GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10
	X	**X**		**X**	**X**		**X**	**X**	

5. **Teaching and learning methods**:

The course will be delivered through a combination of lectures, laboratories and assignments.
Use will be made of the blackboard online learning environment as well as the internet.

6. **Student workload**:

Lectures: 2 x 15 = 30 hours

Laboratories (tutorials): 2 x 15 = 30 hours

Expected minimum independent study: 4 x 15 = 60 hours

Total = 120 hours per semester of student effort

7. **Assessment**

Assignments:

Assignment 1 – Multithreading and synchronization (5%)

Assignment 2 and Assignment 3 are *project based* and the same project is used for them using different technologies – Assignment 2 Client/Server using sockets + multithreading for the server and Assignment 3 using RMI.

Assignments 2 and 3 conclude with a end-of-semester presentation of the students findings and experiences with both the technologies that they used in Assignments 2 and 3 to implement the same project.

Assignments 2 and 3 are worth 7.5% each with remaining 5% for the presentation.

Examinations:

1. Assignments: 25%
2. Mid semester: 20%
3. In class quizzes: 5% (3 quizzes of *5 marks* each. Total quiz marks are downscaled at the end of semester to 5%)
4. End of semester final exam: 50%

8. **Schedule of topics in order will include:**

Wk	Lecture	Topic	Assignment out/in
1.	**0**	Course introduction / *Java Programming Review – not assessed*	
2.	**1**	Introduction to Concurrency – Java Threads	
3.	**2**	Race conditions and synchronisation	
4.	**3**	Condition Synchronisation- classical problems	*A1 out*
5.	**4**	Semaphores and other synchronisation constructs	
6.	**5**	Problems with synchronisation – Deadlocks, livelocks and starvation	
7.	**6**	Verification using FSP	
8.		***Mid Semester Exam***	
9.	**7**	Concurrent Architectures	

10.	**8**	Distributed Programming	*A2 and A3* out
11.	**9**	Message Passing	
12.	**10**	Atomic Operations and Lock free algorithms	
13.	**11**	Revision lecture	
14.			
15.		***Lab Exam***	

9. **Facilities required**:

Specific facilities required

- PC computers
- Windows7/ Linux with JDK 6 or later, DrJava or Eclipse

10 **Reference texts required for library:**

1. *Concurrency: State Models and Java™ Programs, 2e*
Jeff Magee & Jeff Kramer
Wiley, 2006
ISBN-10: 0470093552 ISBN-13: 978-0470093559

2. *Concurrent Programming in Java™: Design Principles and Patterns, 2e*
Doug Lea
Prentice Hall, 1999
ISBN-10: 0201310090 ISBN-13: 978-0201310092

3. *Java™ Concurrency in Practice*
Brian Goetz et al
Addison-Wesley Professional, 2006
ISBN-10: 0321349601 ISBN-13: 978-0321349606

4. *The Java™ Programming Language*, *4e*
Ken Arnold, James Gosling, David Holmes
Prentice Hall, 2006
ISBN-10: 0-321-34980-6 ISBN-13: 978-0-321-34980-4

Online references:
1. *The Java Tutorial* **-** http://docs.oracle.com/javase/tutorial/
(also available for download/offline viewing from the above hyperlink)

Course materials:
Lectures, exercises, tests and examination papers. Some material given as required.

11 **System course coordinator:**
Name: Tauseef Kamal
College: Sohar College of Applied Sciences
Contact details: tauseefk.soh@cas.edu.om

SFDV3006 Concurrent Programming

Coverage of Graduate Attributes

GA2 – *All lectures* cover specialized terminology of concurrent programming in English. Each lectures adds more to the terminology as the course progresses

GA3, GA5 and GA6 – *Coverage: All lectures*. Problem solving exercises for each lecture, assignments and the project. Students need to think for themselves to solve problems presented every week.

GA8 and GA9 – *Coverage: All lectures*. Students need to write programs for all lecture exercises. Well-crafted lab exercises are given corresponding to every lecture which needs to student to program in Java™ as well use various concurrent programming constructs to solve the given problems, lab exercises and assignments.

GA9 – *Coverage: All lectures*. Students get familiar and use Java™ (especially parts of Java™ language related to concurrency), concurrent constructs usage and implementation, implementing message passing, concurrent architectures, modeling and verification using the FSP language, network programming using sockets and distributed programming using Java™ Remote Method Invocation (RMI)