1.	**Name of Course**					**Parallel and Distributed computing**
2.	**Course Code**					**CCPS3623**
3.	**Name(s) of academic staff**					
4.	**Rationale for the inclusion of the course/module in the programme**					**Major** Most modern computer systems are now distributed in nature and parallel by necessity. This module provides an integrated view of the various aspects of the distributed and parallel computing and the various techniques that can be used to program them. It provides students with contemporary knowledge in the topic and the foundations for using the technology in real world applications.
5.	**Semester and Year offered**					2/3
6.	**Total Student Learning Time (SLT)**	**Face to Face**				**Total Guided and Independent Learning**
	L = Lecture T = Tutorial P = Practical O= Others	**L** **28**	**T**	**P** **28**	**O**	Independent=84 Total=140
7.	**Credit Value**					3
8.	**Prerequisite (if any)**					CCPS1563 Object-Oriented Programming
9.	**Objectives**: • To provide students with contemporary knowledge in parallel and distributed computing; • To equip students with skills to design, analyze and implement parallel and distributed applications.					
10.	**Learning outcomes:** Upon the successful completion of this module a student should be able to: • Explain the evolution and fundamental concepts of Concurrent and Distributed systems. • Analyse and Apply Concurrent and distributed algorithms in problem solving. • Develop parallel applications from Message Passing Interface (MPI) programming and Shared Memory programs and distributed system using some programming languages.					
11.	**Transferable Skills:** • Problem-solving skills • Master skills to measure the performance of parallel and distributed programs • Communicate effectively about parallel and distributed systems, with specialists and non-specialists. • Establish a practical basis for the utilization of concurrent and distributed systems.					

12. **Teaching-learning and assessment strategy**

A variety of teaching and learning strategies are used throughout the course, including:

- Classroom lessons. Lectures and Power Point presentations
- Laboratory sessions: Practice exercises
- brainstorming;
- student-Lecturer discussion
- collaborative and co-operative learning;
- Independent study.

Assessment strategies include the following:

- Ongoing quizzes
- Midterm tests
- Performance Assessment (project, Assigned exercises)
- Lecturer Observation

13. **Synopsis:**

The fields of Parallel and Distributed Computing have converged gradually into a single field over the last decade. Parallel and Distributed Computing can be considered an umbrella that incorporates a wide range of theoretical ideas, technologies, and applications. This module presents the fundamental concepts of Parallel and Distributed systems, and the various techniques that can be used to program them.

14. **Mode of Delivery:**

- Classroom lessons. Lectures and Power Point presentations
- Laboratory sessions: Practice exercises

15. **Assessment Methods and Types:**

The assessment for this course will be based on the following:

Coursework **50%**

1. Quizzes and Assignments 15%
2. Project 15%
3. Mid-Semester Exam 20%

Final Examination **50%**

100%

16. **Mapping of the course/module to the Programme Aims**

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
3	4	2	1	2	2	3	3	0	4	3	0

17. **Mapping of the course/module to the Programme Learning Outcomes**

LO1	LO2	LO3	LO4	LO5	LO6	LO7	LO8	LO9	LO10	LO11	LO12
3	4	2	2	2	2	2	2	2	1	2	1

18. **Content outline of the course/module and the SLT per topic**

	Details	SLT			
		L	P	Indep.	Total

Topic 1	**Introduction** Introduction to Concurrent Systems, high performance computing (HPC), Introduction to Synchronization, Programming Notation for specifying concurrent execution Fine-grained and Course-grained Atomic Actions, Strategy for solving Concurrent Problems Introduction to Distributed System, Issues in Distributed Systems, Types of Distributed Systems \ Distributed Systems versus Parallel Systems	4	2	10	16
Topic 2	**Parallel programming:** Fundamentals, bag of tasks paradigm	4	4	12	20
Topic 3	**Semaphores** Informal Definition of a Semaphore Problems with Busy-wait, the method of passing the baton, scheduling, Producers and Consumers, Bounded Buffer, Dining Philosophers, Parallel scientific computing, Barrier synchronization, Multiprocessor implementations	4	4	12	20
Topic 4	**Monitors** Monitors: basic concepts, synchronization techniques Monitors as Objects with Synchronization, Examples such as Shortest Job Next, ReadWrite Controller, The Sleeping Barber.	2	4	8	14
Topic 5	**Threads** Introduction to Threads, Threads in Java, Streams, Threads and Synchronization, Producer/Consumer Problem	2	2	6	10
Topic 6	**TCP/IP Based Distributed System Services** Communication Protocols, The Internet, TCP/IP Protocol UDP, TCP/IP Family of Services Remote operations: tools, RPC and rendezvous Implementation; Java RMI	4	4	12	20
Topic 7	**Communication Paradigms** Communication Paradigms and Network Programming Message Passing Paradigm: basic concepts and examples Message passing: clients and servers Message passing: interacting peers Message passing in Java, MPD, and MPI Remote Procedure Call and RMI TCP and UDP Networking in Java Network Servers C/S Systems	4	6	14	24
Topic 8	**Distributed Computing** Distributed Computing with CORBA Rationale for using CORBA Introduction to main features of CORBA Architecture Developing a client/server based CORBA application Limitations of /Extensions to CORBA Competitors to CORBA	4	2	10	16

		Total	28	28	84	140
	Laboratory	Laboratory Details Exercises based on topics covered in each lecture. Laboratory work must include the following • Parallel Programming: Introductory examples • The compilation process • The linking process • Loading the program • Creating Multitasks • Communication between multitasks • Root debugging • Managing the stack and heap • Detecting Deadlocks • Measuring performance • Distributed Computing • CORBA: Introductory examples • Developing a client/server based CORBA application				
19.	**Main references supporting the course:** **1.** A.D. Kshemkalyani, M. Singhal, **Distributed Computing: Principles, Algorithms, and Systems**, Cambridge University Press, 2008 **2.** Calvin Lin, Larry Snyder, **Principles of Parallel Programming,** Addison-Wesley,2008					
	Additional references supporting the course: 1. Andrew S. Tannenbaum and Maarten van Steen, **Distributed Systems: Principles and Paradigms**, 2nd Edition, Prentice Hall, 2006 2. T. J. Fountain , **Parallel Computing: Principles and Practice**, Cambridge University Press, 2006 3. Vijay K. Garg ,**Concurrent and Distributed Computing in Java**, 1st edition, Wiley-IEEE Press; 2004					
20.	**Other additional information** All materials will be available to the students online.					