COURSE DESCRIPTION FORM

INSTITUTION National University of Computer and Emerging Sciences

PROGRAM (S) TO BE EVALUATED Computer Science

A. Course Description

(Fill out the following table for each course in your computer science curriculum. A filled out form should not be more than 2-3 pages.)

Course Code	CS-3006
Course Title	Parallel and Distributed Computing
Credit Hours	3+0
Prerequisites by Course(s) and Topics	Operating Systems
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	3 Assignments = 5% , 3 Quiz (n-1) = 5%, Projects =10% Midterm-1= 15% , Midterm-2= 15%, Final Exam= 50% • Late submission of assignments will only be awarded a maximum of 50% weightage of the respective assessment. • Plagiarism punishment = zero marks in respective assessment.
Course Coordinator	Dr. Nausheen Shoaib
URL (if any)	-
Current Catalog Description	Asynchronous/synchronous computation/communication, concurrency control, fault tolerance, GPU architecture and programming, heterogeneity, interconnection topologies, load balancing, memory consistency model, memory hierarchies, Message passing interface (MPI), MIMD/SIMD, multithreaded programming, parallel algorithms & architectures, parallel I/O, performance analysis and tuning, power, programming models (data parallel, task parallel, process-centric, shared/distributed memory), scalability and performance studies, scheduling, storage systems, synchronization, and tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE).

Textbook (or **Laboratory Manual** for Laboratory Courses)	**Book#1:Introduction to Parallel Computing, Second Edition** Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar **Book#2: Programming Massively Parallel Processors** By David B. Kirk
Reference Material	**Book#3: Big Data Systems: A 360 degree Approach** By Jawwad Shamsi

Course Goals

A. Course Learning Outcomes (CLOs)

1. Learn about parallel and distributed computers.
2. Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library.
3. Analyze complex problems with shared memory programming with openMP.

B. Program Learning Outcomes

For each attribute below, indicate whether this attribute is covered in this course or not. Leave the cell blank if the enablement is little or non-existent.

1. Academic Education:	To prepare graduates as computing professionals	✔
2. Knowledge for Solving Computing Problems:	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.	✔
3. Problem Analysis:	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.	✔
4. Design/ Development of Solutions:	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	✔
5. Modern Tool Usage:	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.	✔
6. Individual and Team Work:	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.	✔
7. Communication:	Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.	✔

8. Computing Professionalism and Society:	Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.	✔
9. Ethics:	Understand and commit to professional ethics, responsibilities, and norms of professional computing practice.	✔
10. Life-long Learning:	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.	✔

C. Relation between CLOs and PLOs
(CLO: Course Learning Outcome, PLOs: Program Learning Outcomes)

		PLOs									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CLOs	1	✔	✔	✔							
	2	✔	✔	✔							
	3	✔	✔	✔							
	4										
	5										
	6										
	7										

Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)

	Topics to be covered:			
Week #	**List of Topics**	**No. of Weeks**	**Contact Hours**	**CLO**
1	**Introduction**: Introduction to parallel computing**,** Flynn's Taxonomy, Granularity: fine and coarse grained, some general terms parallel execution, parallel overhead, scalability, Shared Memory paradigm, Distributed memory paradigm,Hybrid – shared and distributed memory.	**1**	**3**	**1**

	2,3	**Parallel Programming Platforms:** Scope of Parallelism, Implicit Parallelism: Trends in Microprocessor Architectures, Pipelining and Superscalar Execution, Superscalar Execution: Issue Mechanisms , Superscalar Execution: Efficiency Considerations, Very Long Instruction Word (VLIW) Processors, Limitations of Memory System Performance, Memory System Performance: Bandwidth and Latency, Improving Effective Memory Latency Using Caches, Impact of Caches , impact of bandwidth, Alternate Approaches for Hiding Memory Latency, Multithreading and prefetching, Interconnection Networks for Parallel Computers, static and dynamic network, Network Topologies: Buses and Crossbar. **(Book : Book#1)**	**1.5**	**3**	**1**
	3,4	**Principles of Parallel Algorithm Design:** Preliminaries: Decomposition, Tasks, and Dependency Graphs, Multiplying a Dense Matrix with a Vector, Database Query Processing, Granularity of Task Decompositions, Degree of Concurrency, Critical Path Length, Limits on Parallel Performance, Task Interaction Graphs, sparse matrix example, Processes and Mapping, Decomposition Techniques: recursive decomposition, Array example, data decomposition, matrix example, itemset frequencies, exploratory decomposition, 15-puzzle example,speculative decomposition, simulation of network nodes. Parallel Algorithm Models: Data Parallel Model, Task Graph Model,	**1.5**	**3**	**1,2,3**

		Master-Slave Model, Pipeline / Producer-Comsumer Model, Hybrid Models. **(Book : Book#1)**			
	5	**Programming Shared Address Space (OpenMP):** OpenMP programming model, parallel directive, reduction clause, for loop, nowait clause, scheduling clause: static, dynamic, guided, Data sharing attribute clauses: shared, private, default, reduction, **(Book : Book#1)** **Lab1:** Parallel for with private and shared variables, no wait, Scheduling clause (static and dynamic), barrier , critical and reduction clause, Independent parallel task using sections. **Quiz#1 :** (Parallel Programming Platforms , Principles of Parallel Algorithm Design) – CLO 1 **(18th Sept,23-22nd Sept,23)**	**1**	**6**	**1,3**
	6	**Mid 1 Exams**			
	7	**Programming Shared Address Space (OpenMP):** Synchronization clauses: critical, atomic, barrier, ordered. Parallel For Loops, sections directive, OpenMP Library Functions: int omp_get_num_threads ( ); int omp_get_max_threads ( ); int omp_get_thread_num ( ), **Lab2:** Implementation of parallel algorithms such as Fibonacci series, sorting algorithms etc. **(2nd Oct,23 – 6th Oct,23)**	**1**	**6**	**1,3**

	8	**Programming Using the Message Passing Paradigm:** Principles of MPI, The Building Blocks: Send and Receive Operations, Buffered and non buffered MPI, MPI interface, Starting and Terminating the MPI Library, Communicators, Querying Information, Sending and Receiving Messages, overlapping communication with computation. **Lab3:** MPI installation, Communication rank and size in MPI, MPI_send / MPI_Recv, MPI_status, MPI_Tag. **(Book : Book#1)** **(9th Oct,23 – 13th Oct,23)**	**1**	**6**	**1,2**
	9	**Programming Using the Message Passing Paradigm:** collective communication and computation opertaions: barrier, broadcast, reduction, prefix, scatter, gather. **Lab4:** MPI Scatter, Gather, Bcast, MPI_wait, MPI_Test, MPI_Allgather. **(Book : Book#1)** **(16th Oct,23-20th Oct,23)**	**1**	**6**	**1,2**
	10	**The Hadoop Distributed File System** **Paper: Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1-10. Ieee, 2010.)** **(Ref Book: Book#3)** **Assignment #3** **Quiz #2 :** (Programming Using the Message Passing Paradigm) - CLO 2 **(23rd Oct,23 – 27th Oct,23)**	**1**	**3**	**1**
	11	**Mid II Exams**			
	12	**The Hadoop Distributed File System**	**0.5**	**3**	**1**

		(Ref Book: Book#3) **Paper: Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1-10. Ieee, 2010.)** **PROJECT proposal submissions (6th Nov,23 – 10th Nov,23)**			
	12,13	**Map Reduce Framework:** Map, Partition, shuffle, sort, reduce. Example: word count, URL access count, reverse web link graph, inverted index, stock summary **(Ref Book: Book#3)**	**1.5**	**6**	**1**
	14	**Introduction to Data Parallelism and CUDA C:** Data Parallelism, CUDA program structure, Vector Addition kernel, Device global memory and data transfer, kernel functions and threading. **(Book:Book#2)** **Quiz#3 :** (Map Reduce) - CLO 1 **(20th Nov,23 – 24th Nov,23)**	**1**	**3**	**1,2,3**
	15	**Data-Parallel Execution Model:** Cuda thread organization, mapping threads to multidimensional, synchronization and transparent scalability, assigning resources to blocks, query device properties. **(Book: Book#2)**	**1**	**6**	**1,2,3**
	16	**Project Evaluations**			
		Total	**16**		
Laboratory Projects/Experiments Done in the Course		**Lab 1:** Parallel for with private and shared variables, no wait, Scheduling clause (static and dynamic), and barrier, Critical and reduction clause, Independent parallel task using sections. **Lab 2:** Implementation of parallel algorithms such as Fibonacci series, sorting algorithms etc.			

	Lab 3: MPI installation, Communication rank and size in MPI, MPI_send / MPI_Recv, MPI_status, MPI_Tag. **Lab 4:** MPI Scatter, Gather, Bcast, MPI_wait, MPI_Test, MPI_Allgather.			
Programming Assignments Done in the Course	**Assignment # 1:** Lab1 + Lab 2. **(CLO 3)** **Assignment#2:** Lab 3 + Lab 4 **(CLO 2)** **Assignment #3:** Hadoop installation on a single node (screen shot submission of jps) on vmware, Example running in Map Reduce Framework. **(CLO 1)**			
Class Time Spent on (in credit hours)	**Theory**	**Problem Analysis**	**Solution Design**	**Social and Ethical Issues**
	40%	25%	25%	10%
Oral and Written Communications	Every student is required to submit a project along with its report of not more than 8 pages.			

Instructor Name: Dr. Nausheen Shoaib

Instructor Signature

Date: 22nd August, 2023