# 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 |
| **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.) | Midterms 30%<br>Assignments (Quiz optional) 10%<br>Projects 10%<br>Final Exam 50% |
| **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**<br>Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar<br><br>**Book#2: Programming Massively Parallel Processors**<br>By David B. Kirk |
| **Reference Material** | **Book#3: Big Data Systems: A 360 degree Approach**<br>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)

**1. Topics to be covered:**

| List of Topics | No. of Weeks | Contact Hours | CLO |
|---|---|---|---|
| **Introduction:** 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** |

| | | | | | |
|---|---|---|---|---|---|
| | **( Ref Book: Book#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** | **3** | **1** | |
| | **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 | **1** | **3** | **1,2,3** | |

| | | | | | |
|---|---|---|---|---|---|
| | frequencies, exploratory decomposition, 15-puzzle example,speculative decomposition, simulation of network nodes. Parallel Algorithm Models: Data Parallel Model, Task Graph Model, Master-Slave Model, Pipeline / Producer-Comsumer Model, Hybrid Models. **( Book : Book#1)** | | | | |
| | **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, 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 ( ), **( Book : Book#1))** | **2** | **6** | **1,3** | |
| | **Mid 1 Exams** | | | | |
| | **Programming Using the Message Passing Paradigm:** Principles of MPI, The Building Blocks: Send and Receive Operations, Buffered and non bufferedMP, MPI interface, Starting and Terminating the MPI Library, Communicators, Querying Information, Sending and Receiving Messages, overlapping communication with computation, collective communication and computation opertaions: barrier, broadcast, reduction, prefix, scatter, gather. | **2** | **6** | **1,2** | |

| | | | | |
|---|---|---|---|---|
| | **( Book : Book#1)** | | | |
| | **Distributed File System (HDFS):** Client-server file systems, What is a parallel file system, Google File System (GFS), Design Assumptions and design principles. File System Interface, GFS Master & Chunkservers, files, chunks , Master, Client Interaction Model, Reading and writing to files, namespace, Core Part of Google Cluster Environment, HDFS design goals, architecture, namenode, datanode, rackawareness. Hadoop: Distributed file system. **( Ref Book: Book#3)**<br><br>**(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.)** | **1** | **3** | **1** |
| | **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)** | **2** | **6** | **1** |
| | **Mid 1I Exams** | | | |
| | **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. **CUDA Memories:** Importance of Memory Access Efficiency, CUDA Device Memory Types. **(Book:Book#2)** | **1.5** | **3** | **1,2,3** |
| | **Data-Parallel Execution Model:** Cuda thread organization, | **1.5** | **6** | **1,2,3** |

| | | | | | |
|---|---|---|---|---|---|
| | mapping threads to multidimensional, synchronization and transparent scalability, assigning resources to blocks, query device properties. **(Book: Book#2)** | | | | |
| | Total | **15** | | | |

| | |
|---|---|
| **Laboratory Projects/Experiments Done in the Course** | Lab 1: Parallel Execution in OpenMP<br>Lab 2: Scheduling in OpenMP<br>Lab 3: Critical in OpenMP<br>Lab 4: Independent Parallel Task using Sections in OpenMP<br>Lab 5: Communication rank and size in MPI<br>Lab 6: MPI_send / MPI_Recv<br>Lab 7: MPI Scatter, Gather, Bcast<br>Lab 8: Map Reduce<br>Lab 9: CPU / GPU transfer in CUDA C |
| **Programming Assignments Done in the Course** | **Assignment # 1:**<br>Task : OpenMP Assignment (Lab 1, 2,3,4)<br>**Assignment#2:**<br>Task : MPI Assignment (Lab 5,6,7)<br>**Assignment #3:**<br>Task 1: Hadoop installation on a single node (screen shot submission of jps) on VM<br>Task 2: Map Reduce Assignment (Lab 8)<br>**Assignment#4: (Note: Student list need to be given to IT to make account on GPU server so they can ssh remotely through MobaXterm)**<br>Task: CUDA C Assignment (Lab 9) |

| **Class Time Spent on** (in credit hours) | **Theory** | **Problem Analysis** | **Solution Design** | **Social and Ethical Issues** |
|---|---|---|---|---|
| | 20 | 30 | 40 | 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 07th February, 2022**