

Branch Name:	Computer Engineering (Software Engineering)
Program Code:	ET207
Course Title:	High Performance Computing
Course Code:	2ET2030304T
Pre-requisite Course:	Operating System, Computer Organization and Architecture, Microprocessor and Architecture, Data structure and algorithms.

Course Objectives:

1. To introduce high-performance scientific computing (HPSC) to the postgraduate students in engineering and sciences starting their research in numerical sciences (e.g. computational mechanics, molecular dynamics, etc.).
2. In addition, the course will make significant use of the HPC cluster and helping students get comfortable with the infrastructure for future use in research.
3. This course will cover efficient programming techniques (including Python and shell scripting), version control, parallel programming using OpenMP and MPI, graphics and visualization, and cloud computing.
4. These techniques will help the students carry out their numerical research more effectively. Fortran/C will be used as the base programming language for implementing OpenMP and MPI directives.

Teaching and Examination Scheme:

Credit	Teaching Scheme Hours per week				Evaluation Scheme (Marks)								
					Theory (Marks)				Practical (Marks)				Total (Marks)
	L	T	P	Total	SEE		CCA		SEE		CCA		
					Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
3	3	-	-	3	24	60	16	40	-	-	-	-	100

Course Contents:

Unit No	Topics	Total Hours	Weight age (%)
1	Introduction: Definition of HPC, history and latest developments, Moore's law, introduction to scientific computing, challenges with setting up HPC/data centres (storage, power supply and thermal management), topology of processors, demonstration of the use of HPC to solve a heat conduction problem, sparse matrices, binary storage, fixed-point and floating-point real numbers, IEEE standards, virtual machines, Unix shell and commands.	07	20
2	Introduction to programming languages, version control and makefiles: Introduction to compiled programming languages (precision, compiler optimisation, timing codes, LAPACK and BLAS), Git (Git commands, Github, graphical Git tools, Git demo), build systems and dependency checking.	06	15
3	Cluster Computing and its Architecture: Ease of Computing Scalable Parallel Computer Architecture Towards Low Cost Parallel Computing & Motivation Windows opportunity A Cluster Computer And Its Architecture Cluster Classification Commodity Components for Clusters Network Services/Communication SW Cluster Middleware and Single Systems Image Resource management & Scheduling (RMS)	07	20
4	Introduction to Grid and its Evolution: Introduction to Grid and its Evolution: Beginning of the Grid Building blocks of Grid Grid Application and Grid Middleware Evolution of the Grid: First, Second & Third Generation	04	15

5	Scientific computing: Solution to the steady-state diffusion problem using the finite difference method, Jacobi method with OpenMP and MPI, numerical integrals using Monte Carlo (MC) method, solution to Poisson problem using MC method	06	15
6	Introduction to OpenMP , OpenMP parallelization of Matrix algebra algorithms , Demonstration of OpenMP code , Introduction to MPI , Communications using MPI , Domain decomposition and Schwarz algorithm, Load balancing	07	15

Reference Books:

1. L. R. Scott, T. Clark, B. Bagheri, Scientific Parallel Computing, Princeton University Press, 2005.
2. C. Lin and L. Snyder, Principles of Parallel Programming, Pearson, 2008.
3. LD Fosdick, ER Jessup, CJC Schauble, G Domik An Introduction to High performance Scientific Computing, MIT Pres, 1996.
4. High Performance Cluster Computing, Volume 1, Architecture and Systems, Rajkumar Buyya, Pearson Education.
5. Berman, Fox and Hey, Grid Computing – Making the Global Infrastructure a Reality, Wiley India.
6. R. Chandra, L. Dagum, et. al., Parallel Programming in OpenMP, Academic Press, 2001.
7. M. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill Education, 2003.
8. G. Karniadakis, R. Kirby II, Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2003.
9. W. Gropp, E. Lusk, A. Skjellum, Using MPI, The MIT Press, 2014.

Course Learning Outcomes (CLO):

Course Outcomes	Description	Bloom's Taxonomy Level
COL1	Understand High Performance Computing (HPC) system architectures and various computational models, various programming language	Understanding (2) / Applying (3)
COL2	Understand grid programming concepts and system architectures and basic concepts of cluster	Understanding (2) / Applying (3)
COL3	Apply parallel execution models and methodologies for parallel programming and parallel applications development.	Applying (3)
COL4	Design and implement compute intensive applications on HPC platform.	Applying (3)
COL5	Understanding and applying OpenMP and MPI based parallelization of iterative matrix solvers with practical approach	Understanding (2) / Applying (3)

Mapping of CLO with PO and PSO

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes (PSOs)			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CLO1	H	M	H	H	H			L							H	M
CLO2	H	M	H	H	H			L				L			H	M
CLO3	H	M	M	M	H			L							M	M
CLO4	M	M	M	H	H			L				L			M	M
CLO5	M	M	M	H	M	M		L				L			M	M

H: High, M: Medium, L: Low