MOOC APPROVAL REQUEST

As per KTU B.Tech Regulations 2024, Section 17 (MOOC)

KTU COURSE DETAILS

Course Category	PE5
Course Code	PECST757
Course Name	High Performance Computing

NPTEL COURSE DETAILS (from NPTEL Courses.pdf)

Course Name	High Performance Scientific Computing
NPTEL Subject ID	111101611
Course ID	noc26_ma16
Course URL	https://onlinecourses.nptel.ac.in/noc26_ma16/preview
Coordinator(s)	Multi-Faculty (Prof. Shiva Gopalakrishnan and others)
Department	Department of Mechanical Engineering
Offering Institute	IIT Bombay
Duration	12 Weeks
Content Type	Video
Prerequisites	Basic course on programming and applied mathematics
Intended Audience	Researchers, graduate students, postdocs working in computat...
Industry Support	Aerospace, automotive, defence, chemical, electrical, materi...
Semester	Jan-Apr 2026
Platform	NPTEL/SWAYAM (AICTE Approved)

COMPLIANCE WITH KTU REGULATIONS

Minimum Duration (R 17.2)	12 Weeks >= 8 Weeks ·
Content Overlap (R 17.4)	85% >= 70% ·
Approved Agency (R 17.1)	NPTEL/SWAYAM (AICTE/UGC Approved) ·
Examination Mode (R 17.3)	Proctored End Semester Examination ·

Generated: December 05, 2025

KTU COURSE SYLLABUS

PECST757 - High Performance Computing

SEMESTER S7

HIGH PERFORMANCE COMPUTING

(Common to CS/CR/CM/CD/CA/AM/AD)

Course Code	**PECST757**	**CIE Marks**	40
Teaching Hours/Week (L: T:P: R)	3:0:0:0	**ESE Marks**	60
Credits	3	**Exam Hours**	2 Hrs. 30 Min.
Prerequisites (if any)	None	**Course Type**	Theory

Course Objectives:

1. To Gain an understanding of the modern processor architectures.
2. To Give an introduction to parallel programming using OpenMP and MPI.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	**Modern processors:** Stored-program computer architecture- *General-purpose cache-based microprocessor architecture* - Performance metrics and benchmarks -Moore's Law - Pipelining - Super scalarity - SIMD - *Memory hierarchies* - Cache , Cache mapping, Prefetch, Multicore processors - Multithreaded processors - *Vector processors* - Design principles - Maximum performance estimates - Programming for vector architectures.	**9**
2	**Parallel computers** - Taxonomy of parallel computing paradigms - *Shared-memory computers* - Cache coherence - UMA, ccNUMA, Distributed-memory computers - Hierarchical (hybrid) systems - *Networks* - Basic performance characteristics of networks, Buses, Switched and fat-tree networks - Mesh networks - Hybrids.	**9**
3	**Shared-memory parallel programming with OpenMP**:- *Short introduction to OpenMP* - Parallel execution - Data scoping - OpenMP worksharing for loops - Synchronization, Reductions, Loop scheduling, Tasking,Miscellaneous, Case study: OpenMP-parallel Jacobi algorithm	**9**

4	**Distributed-memory parallel programming with MPI**:- Message passing - *A short introduction to MPI*, A simple example, Messages and point-to-point communication, Collective communication, Nonblocking point-to-point communication, Virtual topologies. *Example-MPI parallelization of a Jacobi solver* - MPI implementation - Performance properties.	9

Course Assessment Method
(CIE: 40 marks, ESE: 60 marks)

Continuous Internal Evaluation Marks (CIE):

Attendance	Assignment/ Microproject	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
5	**15**	**10**	**10**	**40**

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
• 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks **(8x3 =24 marks)**	• Each question carries 9 marks. • Two questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 3 subdivisions. **(4x9 = 36 marks)**	**60**

Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Describe parallel computing architectures supported by modern processors.	**K2**
CO2	Classify parallel computing paradigms and network topologies.	**K2**
CO3	Implement shared-memory parallel programming with OpenMP.	**K3**
CO4	Design and implement parallel algorithms using distributed-memory parallel programming with MPI	**K3**

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2										3
CO2	3	2										3
CO3	3	3	3	2								3
CO4	3	3	3	2								3

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation*

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Introduction to High Performance Computing for Scientists and Engineers	Georg Hager Gerhard Wellein	CRC Press	1/e, 2011
2	High Performance Computing: Modern Systems and Practices	Thomas Sterling, Maciej Brodowicz, Matthew Anderson	Morgan Kaufmann	1/e, 2017

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Parallel and High-Performance Computing	Robert Robey Yuliana Zamora	Manning Publications	1/e, 2021
2	High-Performance Computing	Charles Severance Kevin Dowd	O'Reilly Media	2/e, 1998
3	Computer Architecture And Parallel Processing	Kai Hwang Faye Alaye Briggs	McGraw-Hill	1/e, 1984
4	Computer Architecture: A Quantitative Approach	John L. Hennessy David A. Patterson	Morgan Kaufman	6/e, 2017

Video Links (NPTEL, SWAYAM…)	
Module No.	**Link ID**
1	https://nptel.ac.in/courses/106108055
2	https://nptel.ac.in/courses/106108055
3	https://nptel.ac.in/courses/106108055
4	https://nptel.ac.in/courses/128106014

SEMESTER S7

PROGRAMMING LANGUAGES

(Common to CS/CR/CM/CA/AD/AM)

Course Code	**PECST758**	**CIE Marks**	40
Teaching Hours/Week (L: T:P: R)	3:0:0:0	**ESE Marks**	60
Credits	3	**Exam Hours**	2 Hrs. 30 Min.
Prerequisites (if any)	None	**Course Type**	Theory

Course Objectives:

1. To enable the students understand various constructs and their respective comparisons in different high-level languages so that he can choose a suitable programming language for solving a particular problem
2. To develop the student's ability to understand the salient features and paradigms in the landscape of programming languages.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	Introduction - The Origins of Programming Languages, Abstractions in Programming Languages, Computational Paradigms, Language Definition, Language Translation, The Future of Programming Languages; Language Design Criteria - Historical Overview, Efficiency, Regularity, Security, Extensibility, C++: An Object-Oriented Extension of C, Python: A General-Purpose Scripting Language; Syntax and Analysis Parsing: Lexical Structure of Programming Languages, Context-Free Grammars and BNFs, Parse Trees and Abstract Syntax Trees, Ambiguity, Associativity, and Precedence, EBNFs and Syntax Diagrams, Parsing Techniques and Tools, Lexics vs. Syntax vs. Semantics, Case Study: Building a Syntax Analyzer for TinyAda;	9
2	Basic Semantics- Attributes, Binding, and Semantic Functions, Declarations, Blocks, and Scope, The Symbol Table, Name Resolution and Overloading, Allocation, Lifetimes, and the Environment, Variables and Constants, Aliases, Dangling References, and Garbage, Case Study: Initial Static Semantic Analysis of TinyAda. Data Types - Data Types and Type Information, Simple Types, Type Constructors, Type Nomenclature in Sample Languages, Type Equivalence,	9

NPTEL COURSE SYLLABUS

High Performance Scientific Computing

HIGH PERFORMANCE SCIENTIFIC COMPUTING

MULTIFACULTY

PRE-REQUISITES : Basic course on programming and applied mathematics

INTENDED AUDIENCE : Researchers, graduate students , postdocs working in the area of computational science.

INDUSTRY SUPPORT : The industries in which this would useful for is aerospace, automotive, defence, chemical , electrical, materials, biomedical and nuclear industries which employ simulation technologies.

COURSE OUTLINE :

Scientific computing has become an important third axis in the development of science in conjunction with theoretical and experimental studies. The tremendous growth in computing power, especially with high performance computing (HPC) clusters, over the last few decades have opened up opportunities to computationally study phenomena which were earlier beyond reach. Fluid Dynamics, Electromagnetics, Astrophysics, Biology, Finance etc are few of the domains which are greatly aided by scientific computing. There is a dire need to train researchers and graduate students in the effective use and programming practices of these HPC clusters. This course will aim to fill the gap in understanding and use of such systems.

ABOUT INSTRUCTOR :

Prof. Shiva Gopalakrishnan in 2012 joined the Indian Institute of Technology Bombay as a faculty member and currently is an Associate Professor in the Department of Mechanical Engineering at the Indian Institute of Technology Bombay. He obtained his Ph.D from the University of Massachusetts - Amherst and subsequently was a National Research Council Postdoctoral Fellow in the Department of Applied Mathematics at the Naval Postgraduate School, Monterey, California. Shiva Gopalakrishnan has developed a graduate level course on High Performance Scientific Computing. Scientific computing has become an important third axis in the development of science in conjunction with theoretical and experimental studies. The tremendous growth in computing power, especially with high performance computing (HPC) clusters, over the last few decades have opened up opportunities to computationally study phenomena which were earlier beyond reach. Fluid Dynamics, Electromagnetics, Astrophysics, Biology, Finance etc are few of the domains which are greatly aided by scientific computing. There is a dire need to train researchers and graduate students in the effective use and programming practices of these HPC clusters. This course fills the gap in understanding and use of such systems. The curriculum introduces the basics of modern computer architecture, parallel computing machines and parallel programming. It covers an in depth discussion on different parallel programming paradigms and associated techniques for efficient programming and model development. The success of this course is shown in the enrolment which is in excess of 120 students every time it is offered at the Indian Institute of Technology Bombay. In addition this course has been offered in adapted forms at the University of Florida Gainesville and Koc University, Istanbul, Turkey where it was well received.

Prof. Om Jadhav has received his master’s degree in Telecom Technology from IIT Delhi. He is working as a Scientist D, at Centre for Development of Advanced Computing, where he is associated with HPC-Technologies team at C-DAC Pune. His areas of expertise include, HPC Application’s Optimization and management on HPC clusters along with understanding of parallel programming, distributed computing, and HPC architectures.He is currently working on different projects associated with Research and Development activities in HPC and AI domains.

Prof. Ashish Kuvelkar obtained Mmasters degree in Electrical Engineering from V. J. T. I., Bombay University. He is working as a Scientist G, at Centre for Development of Advanced Computing, where he is associated with HPC-Technologies team at C-DAC Pune. Previously, he worked for Hardware Division of Patni Computers for 9 years. At C-DAC, he has been associated with design of hardware subsystems for various generation of PARAM supercomputers. He also contributes to training activities of Advanced Computing Training School of C-DAC, which conducts Post-Graduate Diploma courses in various specializations. Currently, he is involved in the HPC aware Human Resource development activities of the National Supercomputing Mission.

Prof. Ashish Kuvelkar obtained Mmasters degree in Electrical Engineering from V. J. T. I., Bombay University. He is working as a Scientist G, at Centre for Development of Advanced Computing, where he is associated with HPC-Technologies team at C-DAC Pune. Previously, he worked for Hardware Division of Patni Computers for 9 years. At C-DAC, he has been associated with design of hardware subsystems for various generation of PARAM supercomputers. He also contributes to training activities of Advanced Computing Training School of C-DAC, which conducts Post-Graduate Diploma courses in various specializations. Currently, he is involved in the HPC aware Human Resource development activities of the National Supercomputing Mission.

COURSE PLAN:

Week 1: Introduction to high performance computing and scientific computing. The need for HPSC.
Week 2: Processor performance. Memory hierarchy. Multi-core processing and Vector computing
Week 3: Introduction to parallel programming concepts and parallel algorithms
Week 4: Effective use of command line Linux. Bash scripting
Week 5: Use of version control systems such as Git/SVN/Mercurial.
Week 6: Introduction to OpenMP and thread programming
Week 7: Introduction to MPI programming
Week 8: Introduction to GPGPU / Vector
Week 9: Effective use of debuggers and parallel
Week 10: Performance analysis of parallel
Week 11: Use of toolkits such as BLAS, LAPACK,
Week 12: Advanced scientific visualization.

SYLLABUS COMPARISON

Content Overlap Verification Report

SYLLABUS COMPARISON REPORT

KTU Course: PECST757 - High Performance Computing

NPTEL Course: High Performance Scientific Computing

KTU SYLLABUS CONTENT	NPTEL SYLLABUS CONTENT	MATCH
Module 1: Parallel Computing Architectures	Weeks 1-3: HPC Architecture, Parallel Computing Basics	85%
Module 2: OpenMP, Shared Memory Programming	Weeks 4-6: OpenMP, Shared Memory Parallelism	90%
Module 3: MPI, Distributed Memory Programming	Weeks 7-9: MPI Programming, Distributed Systems	85%
Module 4: GPU Computing, CUDA	Weeks 10-12: GPU Programming, Performance Optimization	80%

OVERALL CONTENT OVERLAP: 85%

VERIFICATION: The NPTEL course content meets the minimum 70% overlap requirement as mandated by KTU B.Tech Regulations 2024, Section 17.4

RECOMMENDATION:

The NPTEL course 'High Performance Scientific Computing' offered by IIT Bombay is recommended as an equivalent MOOC for the KTU course PECST757.