Part I Courses

Module 1.1:	Scientific Programming Environment
Coordinator	Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	This course will introduce Unix-like operating systems, show how to setup the scientific programming environment in such operating systems. It will present the modern software tools required to provide such an environment and discuss important points like documentation and testing.
Main Topics	• Introduction to Unix-like operating systems (kernel vs. userspace, processes/threads, file system semantics) • Shell scripting (bourne shell) • Basics of python programming • Review of basic concepts of C and Fortran programming language, and mixing thereof • Compiler architecture (preprocessor, compiler, assembler, linker) and use of libraries • Build automation tools (make and CMake) • Source code management (git) • Unit and regression testing • Software documentation
Objectives	On successful completion of this module students should have their own software environment and tools prepared and configured for the rest of the activities of the Master Program. Students learn the workflows of software development and working collaboratively. These concepts are the basis on which to develop efficient scientific code for high-performance computing applications.

Module 1.2:	**Introduction to Computer Architectures for HPC**
Coordinator	Stefano Cozzini (CNR-IOM) Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to key topics in computer architecture needed in HPC environment including a detailed overview on parallel architectures.
Main Topics	• Computer architecture • Memory hierarchy • Modern multi-core CPU systems • Overview on massively parallel processors • Parallel architectures for HPC and future trends • Benchmarking and energy efficiency in HPC • Profiling of a serial applications • False sharing and memory efficiency on Multi-core systems • Practical introduction to software optimization
Objectives	On successful completion of this module students should be able to understand basic concepts of modern computer architecture, and know the different kinds of parallel architectures commonly applied to scientific research.

Module 1.3:	**Parallel Programming**
Coordinator	Ivan Girotto (ICTP) Nicola Cavallini (SISSA)

	Alberto Sartori (SISSA)
Module Description	Introduction to key topics in parallel programming. Main parallel programming paradigms: message passing (MPI) and multi-threading (OpenMP).
Main Topics	• Introduction to parallel computing • Principles of parallel algorithm design and multi-level parallelism • Message-passing parallelization (MPI) • Shared memory parallelization (OpenMP) • Analysis of scalability and parallel performance metrics • Overview on the debugging and the profiling of parallel applications • Practical introduction to parallel Math libraries • Further trends of parallel programming in HPC
Objectives	On successful completion of this module students should be able to write parallel programs and know fundamental techniques of parallel programming to develop parallel applications along with methods of parallel performance analysis and debugging.

Module 1.4:	**Introduction to Numerical Analysis**
Coordinator	Luca Heltai (SISSA) Gianluigi Rozza (SISSA)

	Nicola Cavallini (SISSA) Alberto Sartori (SISSA)
Module Description	Introduction to numerical analysis, with focus on linear algebra, polynomial approximation, numerical integration and numerical solution of ODEs
Main Topics	• The foundations of numerical analysis • Polynomial Interpolation • Numerical Integration • Resolution of non-linear systems • Resolution of large linear systems • Eigenvalues approximation • Numerical solution of ODEs • Numerical solution of PDEs
Objectives	On successful completion of this module students should be able to understand and implement in computer program numerical integration, numerical derivation, functional interpolation and basic linear algebra operations.

Module 1.5:	**Object Oriented Programming**
Coordinator	David Grellscheid (University of Durham) Nicola Cavallini (SISSA)

	Alberto Sartori (SISSA)
Module Description	Introduction to fundamental concepts of programming from an object-oriented perspective.
Main Topics	• Classes and objects • Abstraction • Encapsulation • Calling methods and passing parameters • Exceptions • Inheritance • Polymorphic variables and methods • OOP in C++ • C++ Standard Template Library • Design issues • OOP in Python
Objectives	On successful completion of this module students should be able to understand modern software engineering and design principles and should develop fundamental programming skills in the context of a language that supports the object-oriented paradigm.

Module 1.6:	**Parallel Data Management & Data Exchange**
Coordinator	Stefano Cozzini (CNR) Luca Bortolussi (University of Trieste)
Module Description	The module introduces modern techniques to deal with the large amount of data in scientific and technical computing.
Main Topics	• Introduction to Big data issues • Parallel file systems and parallel I/O • Scientific data formats and libraries (NetCDF, HDF5) • MPI-IO • Data intensive computing (distributed file systems and MapReduce), Hadoop • Web interface and protocols for data exchange (i.e. opendap) • Workflows for data processing • Benchmarking and profiling data intensive calculation
Objectives	On successful completion of this module students have an overview of the main techniques and tools to tackle data-intensive computational problems.

Module 1.7:	**Advanced Computer Architectures & Optimizations**
Coordinator	Piotr Luszczek (University of Tennessee, USA) Chris Dahnken (INTEL)
Module Description	The course presents advanced topics in optimization techniques needed in HPC environment. In particular it will focus on the use of application accelerators in high-performance and scientific computing and issues that surround it.
Main Topics	● Advanced optimization techniques ● Memory management and optimization ● Introduction to novel accelerator processors, systems, and architectures ● Introduction to GPU computing ● Overview of accelerated architecture ● Programming interfaces for accelerator ○ CUDA ○ OpenCL ○ OpenACC ● Specific libraries with accelerator support
Objectives	On successful completion of this module, students will have an overview of the advanced computational architectures and accelerators and how to use them.

Module 1.8:	**High Performance Computing Technology**
Coordinator	Moreno Baricevic (CNR-IOM) Stefano Cozzini (CNR-IOM) Stefano Piani (eXact-Lab)
Module Description	This module introduces state-of-the-art technologies and innovation in High Performance Computing. Main components of computing infrastructure are analyzed and discussed. Students will install and configure a HPC Linux Cluster and will also be exposed to the use of Cloud and Grid Infrastructures.
Main Topics	• HPC system deployment • Software Provisioning (modules) • Managing hardware diversity • Scheduling and resource management • Usage accounting • Data management (quotas, purging, archival) • Sustainable HPC computing infrastructure • Green computing • Grid and Cloud Computing
Objectives	On successful completion of this module students should be able to understand common problems related to the installation and maintenance of a sustainable HPC infrastructure.

Module 1.9:	**Best Practice in Scientific Computing**
Coordinator	Axel Kohlmeyer (Temple U. & ICTP)
Module Description	A module where students are introduced to best practices in scientific computing from different perspective: software development with modern software engineering techniques, optimal exploitation of different HPC platforms, usage and maintenance of large scientific software packages.
Main Topics	● Debug versus optimized mode ● Multilanguage programming ● Strategies for developing scientific codes ● Collaborative ways of developing scientific and technical packages. ● Tools for developing large software packages
Objectives	On successful completion of this module students should have a clear idea about the most successful practices that can be adopted in developing technical and scientific software and make them run efficiently on modern HPC platform.