

COURSE OF STUDY *Physics (LM-17)*
ACADEMIC YEAR 2023-2024

ACADEMIC SUBJECT *Advanced Programming in C++*

General information	
Year of the course	1st
Academic calendar (starting and ending date)	2nd semester: March - May 2024
Credits (CFU/ECTS):	3
SSD	FIS/01
Language	English
Mode of attendance	Compulsory

Professor/ Lecturer	
Name and Surname	Francesco Saverio Cafagna
E-mail	Francesco.cafagna@ba.infn.it
Telephone	+39 0805442328
Department and address	INFN, sezione di Bari, via Orabona 4, 70125 Bari
Virtual room	
Office Hours (and modalities: e.g., by appointment, on line, etc.)	From 10:30 up to 11:30; Physics Department, room 132, or by appointment

Work schedule			
Hours			
Total	Lectures	Hands-on (laboratory, workshops, working groups, seminars, field trips)	Out-of-class study hours/ Self-study hours
75	16	15	44
CFU/ECTS			
3	2	1	

Learning Objectives	<p>The course focuses on the foundations of programming, introducing the students to the evolution of the programming paradigms, from procedural to the object orientation programming styles. Leveraging the characteristics of the C++ programming language, that is its general purpose nature, the granular memory control and to the intrinsic strongly type orientation, will be used as an archetype for such an evolution; for this reason the basic grammar of the programming language will be treated along with all the features that make it the language of choice for object oriented programming.</p> <p>Besides the C++ grammar, an overview of the latest features, introduced with the language's recent release will be given.</p> <p>The course agenda includes exercises designed to examine in depth the theoretical argument treated during the lesson.</p>
Course prerequisites	<p>Knowledge of a personal computer.</p> <p>Knowledge of the most used operating systems like: Linux/Unix, Windows and MAC OS.</p>

Teaching strategie	<p><i>Lectures with hands on exercises during the lecture. The teacher either will explain exercises while students will be able to reproduce them on their personal computer or in a virtual programming environment embedded in an online e-learning platform.</i></p>
---------------------------	--

Expected learning outcomes in terms of	
Knowledge and understanding on:	<ul style="list-style-type: none"> o Learning how to program using procedures. o Learning how to program using object orientation. o Understanding how to design and implement software custom tailored to specific problems related to data analysis or detector read-out and monitor.
Applying knowledge and understanding on:	<ul style="list-style-type: none"> o Ability in development of any kind of algorithm on any computing platform for any kind of application.
Soft skills	<ul style="list-style-type: none"> • <i>Making informed judgments and choices</i> <ul style="list-style-type: none"> o To select the best design solution for a given original problem to be solved using custom made programs. o To select the best programming language feature in terms of efficiency and resource usage. o To select the best library tool to be used to implement a given algorithm. • <i>Communicating knowledge and understanding</i> <ul style="list-style-type: none"> o Understand the semantics and jargon of a programming language. o Learn how to efficiently describe a code using appropriate terminology and information technology concepts. • <i>Capacities to continue learning</i> <ul style="list-style-type: none"> o Learn how to find and use updated references on a programming language, new features and libraries. o Learn how to create, maintain, scale and update original code capable of solving a given research problem.
Syllabus	
Content knowledge	<p><i>Introduction:</i></p> <ul style="list-style-type: none"> - <i>From procedural programming to object oriented programming style: an overview.</i> - <i>Programming: a general introduction.</i> - <i>Programming: overview of base concepts.</i> - <i>Programming: the jargon.</i> - <i>Introduction to the basic usage of the programming tool.</i> - <i>An object oriented programming language: C++.</i> <p><i>The C++ base grammar:</i></p> <ul style="list-style-type: none"> - <i>Base types.</i> - <i>Expressions and statements.</i> - <i>Functions.</i> - <i>Pointer and reference.</i> - <i>Exercises and examples on base grammar.</i> <p><i>C++ advanced features:</i></p> <ul style="list-style-type: none"> - <i>Function overloading.</i> - <i>Namespaces.</i> - <i>Template programming</i> - <i>Structure</i> - <i>Exercise and examples on advanced features.</i> <p><i>Class:</i></p> <ul style="list-style-type: none"> - <i>An introduction and general properties.</i> - <i>Class members: creators, destructor, methods and helper functions.</i> - <i>Member access control: public and private.</i> - <i>Operators and member function overloading.</i> - <i>Exercise and examples in class.</i> <p><i>Class, advanced features:</i></p> <ul style="list-style-type: none"> - <i>Base and Derived class.</i> - <i>Inheritance and polymorphism.</i> - <i>Exercises and examples on class advanced features.</i> <p><i>The Standard Template Library (STL):</i></p>

	<ul style="list-style-type: none"> - <i>Introduction and general features.</i> - <i>Overview of the most common types: string, containers, functionals and algorithms.</i>
Texts and readings	<p><i>B. Stroustrup, The C++ programming language (Third edition), Addison – Wesley.</i></p> <p><i>B. Stroustrup, Programming -- Principles and Practice Using C++, Addison -Wesley ISBN 978-0321543721. December 2008.</i></p> <p><i>S. Lippman et al., C++ Primer (Fifth edition), Addison – Wesley.</i></p> <p><i>S. Meyer, Effective C++ Third Edition, Addison-Wesley ISBN-13: 978-0321334879 (any edition of this book is worth consulting).</i></p> <p><i>S. Meyer, Effective Modern C++, O'Reilly Media, Incorporated; ISBN-13: 978-1491903995.</i></p> <p><i>H. Sutter, Exceptional C++: 47 engineering Puzzles, Programming Problems and Solutions. Reading, MA: Addison-Wesley.</i></p>
Notes, additional materials	
Repository	https://baltig.infn.it/fc-courses/it/programming/cpp-mag-2025

Assessment	
Assessment methods	<i>Written test focused on the implementation of an original code that leverages the usage of object oriented style, and an oral test based on the discussion of the code details.</i>
Assessment criteria	<ul style="list-style-type: none"> ● <i>Knowledge and understanding</i> <ul style="list-style-type: none"> ○ Evaluation of the knowledge of the basic grammar of the programming language. ● <i>Applying knowledge and understanding</i> <ul style="list-style-type: none"> ○ To the ability of applying different programming techniques to problem solving. ● <i>Autonomy of judgment</i> <ul style="list-style-type: none"> ○ To be able to choose between different language semantics and features and to evaluate the impact in an actual case study. ● <i>Communicating knowledge and understanding</i> <ul style="list-style-type: none"> ○ To be able to describe in a clear and synthetic way a code, the design choices applied and the behavior of the various elements integrated in the code. ● <i>Communication skills</i> <ul style="list-style-type: none"> ○ Be acquainted with the programming language and Information Technology jargon and be able to use an appropriate dictionary to express in a clear and shareable way a code or a full project. ● <i>Capacities to continue learning</i> <ul style="list-style-type: none"> ○ Be acquainted with the update policy of a programming language steering committee and the resources to be used for a continuous update on the language, and related libraries, evolution.
Final exam and grading criteria	<i>The clarity in a code explanation and in an argument presentation will be the leading criteria. The right usage of terms in the description of the programming language syntax and library components used in a code. The grade of autonomous judgment in the choice of the design solution to be implemented.</i>
Further information	
	.