YACHAY TECH UNIVERSITY

COURSE PROGRAM

1. General Information

A.	**SCHOOL**	Physical Sciences and Nanotechnology			**B.**	**MAJOR**	Physics Nanotechnology
C.	**COURSE**	Computational Physics II			**D.**	**CODE**	PHYSEDH05 PHYSEDN22
E.	**CURRICULAR UNIT**	Professional			**F.**	**MODALITY**	Face to face
G.	**TOTAL HOURS**	48 [1]	48 [2]	104 [3]	**H.**	**SEMESTER**	8th

2. Prerequisites and Corequisites

PREREQUISITES		COREQUISITES	
COURSES	**Code**	**COURSES**	**Code**

3. Course Description

This is an advanced course on object-oriented programming for physics. It is the second module of the computational physics series taught at Yachay Tech. The course focuses on introducing advanced numerical methods and simulation techniques used in physics, and provides an overview of recent progress made in several areas of scientific computing. The course includes detailed step-by-step examples of how to design software and use parallel programming to solve problems in physics. Topics range from advanced data analysis, through ordinary and partial differential equations, nonlinear dynamics and chaos, basic thermodynamics and fluid simulations, to an introduction to machine learning. Each section of the course includes practical examples from different areas of science and technology in which computational physics and high performance computing have played an important role in recent years.

4. Course Contribution to professional training

This course helps students improve programming skills for the design and implementation of parallel software dedicated to applications in physics.

[1] Teaching Hours. For courses with NON-VALID curriculums, take into account the total number of hours of each course found in each curriculum and place it in this space

[2] Hours of Internship and Experimental Learning

[3] Hours of Independent Learning

5. Course objectives

- Improve object-oriented programming skills to solve physics problems within Linux environments.
- Design algorithms, implement and debug parallel software within the message passing interface in Python language.
- Apply numerical methods and computational techniques to simulate physical systems via reusable and extensible code packages.
- Use high-performance computing in research applications on thermodynamics, fluid dynamics, astrophysics, electromagnetism, particle physics, classical mechanics, quantum mechanics, and other areas of physics.

6. Units / Contents

CURRICULAR UNITS	CONTENTS
UC.1 Ordinary differential equations in physics	Ordinary differential equations, and initial value problems
	Euler methods, Runge-Kutta methods, and applications
	Boundary value problems, shooting and finite difference methods, applications
UC.2 Partial differential equations in physics	Partial differential equations, generalities and classification
	Methods of solving partial differential equations
	Applications to electromagnetism, heat flow, and quantum mechanics
UC.3 Computational fluid dynamics	Discretisation, meshing and conservation in computational fluid dynamics
	Advection, shocks and solitons
	Introduction to hydrodynamics and computational fluid dynamics (CFD) applications
UC.4 Special topics in computational physics	Thermodynamic simulations and introduction to molecular dynamics
	Nonlinear dynamics, chaotic systems, fractals and statistical growth
	Introduction to machine learning
UC.5 Software design and parallel computing for physics	Software design using object-oriented programming
	Message Passing Interface (MPI) and parallel computing
	High Performance Computing (HPC)

7. Learning outcomes of the courseImprove object-oriented programming skills to solve physics problems within Linux environments.

A.	Improve object-oriented programming skills to solve physics problems within Linux environments.
B.	Design algorithms, implement and debug parallel software within the message passing interface in Python language.
C.	Apply numerical methods and computational techniques to simulate physical systems via reusable and extensible code packages.
D.	Use high-performance computing in research applications on thermodynamics, fluid dynamics, astrophysics, electromagnetism, particle physics, classical mechanics, quantum mechanics, and other areas of physics.

8. Methodology

1. Interactive lectures including theory and programming tasks.
2. Laboratory classwork including programming exercises and quizzes on reading material.
3. Individual and group projects including programming homework and research.

9. Information Sources (Bibliography)

9.1 Main

Author/s	Title of Work	Edition	Year of Publication	Publishing house - Country	Availability at YACHAY TECH Library
Landau, Rubin	Computational physics : problem solving with python	3rd	2015	Wiley-VCH; John Wiley - Germany	530.0113 L2539c 2015

9.2 Complementary

Author/s	Title of Work	Edition	Year of Publication	Publishing house - Country	Availability at YACHAY TECH Library
Pang, Tao	An introduction to computational physics	2nd	2006	Cambridge University Press – United States	530.0285 P1917a 2006
Kong, Qingkai Siauw, Timmy Bayen, Alexandre	Python Programming and Numerical Methods - A Guide for Engineers and Scientists	1st	2020	https://pythonnumericalmethods.berkeley.edu	Online

10. Student's Evaluation

Midterm Exam (MT)	☑	Formative Evaluation (FO)	☑	Laboratory (LAB)	☑	Final Exam (FI)	☑

Based on the Academic Regime Regulation issued by the Higher Education Council (CES in Spanish) and the Academic Regime Regulation of Yachay Tech
The inputs that contribute to the completion of this format must be taken from the major project approved by CES.

Prepared by:	Reviewed by:	Approved by:
PROFESSOR - PROFESSORS	MAJOR COORDINATOR - MAJOR DIRECTOR	DEAN – DIRECTOR
SIGNATURE AND DATE:	SIGNATURE AND DATE:	SIGNATURE AND DATE: