YACHAY TECH UNIVERSITY

COURSE PROGRAM

1. General Information							
A.	SCHOOL	Physical Sciences and Nanotechnology			B.	MAJOR	Physics
C.	COURSE	Computational Physics II			D.	CODE	ECFN-PHY-1109
E.	CURRICULAR UNIT	Professional			F.	STUDY MODE	In person
G.	TOTAL HOURS	48 [1]	48 [2]	48 [3]			

2. Prerequisites and Corequisites			
PREREQUISITES		COREQUISITES	
COURSE	Code	COURSE	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
[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
Unit 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
Unit 2. Software design using object-oriented programming	Python classes and modules
	Standalone modules and python packaging for physics
	Timing tests and efficient coding
Unit 3. Parallel computing for physics	High-performance computing (HPC) and job managers
	Parallel computing, CPU/GPU, multiprocessing and joblib
	The Message Passage Interface (MPI) and Application Programming Interfaces (APIs)
Unit 4. Partial differential equations in physics	Partial differential equations, generalities, classification and solving methods
	Parabolic and elliptical problems. Heat and Poisson equations
	Hyperbolic problems: Introduction to computational fluid dynamics (CFD)
Unit 5. Special topics in computational physics	Advection, thermodynamic simulations and introduction to molecular dynamics
	Nonlinear dynamics, chaotic systems, fractals and statistical growth
	Introduction to machine learning

7. Learning outcomes of the course

CAREER LEARNING OUTCOMES
(ONLY THOSE THAT SPECIFICALLY APPLY TO THE SUBJECT)

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	Elsevier: https://pythonnumericalmethods.berkeley.edu/notebooks/Index.html	Online

10. Student's Evaluation			
10.1. Evaluation during the course			
Midterm Exam (EME) ☑	Formative Evaluation (EFO) ☑	Laboratory (LAB) ☑	Final Exam (EFI) ☑

The inputs that contribute to the completion of this format must be taken from the major project or major redesign approved by CES.

Prepared by:	Reviewed by:	Approved by:
PROFESSOR - PROFESSORS	DESIGNATED PERSONNEL	DEAN – LANGUAGE DIRECTOR
SIGNATURE AND DATE:	SIGNATURE AND DATE:	SIGNATURE AND DATE: