UNIVERSIDAD YACHAY TECH

YACHAY TECH UNIVERSITY

SYLLABUS

1. General Information					
A.	SCHOOL	Physical Sciences and Nanotechnology	B.	MAJOR	Physics
C.	COURSE	Computational Physics II	D.	CODE	ECFN-PHY-1109
E.	LEVEL	N/A	F.	ACADEMIC TERM	ISEM2025
G.	CURRICULAR UNIT	Professional	H.	STUDY MODE	In person
I.	TOTAL HOURS	144	J.	PROFESSORS	Wladimir Eduardo Banda Barragán
K.	WEEKLY CLASS SCHEDULE	15:00 - 17:00 Tuesday 13:00 - 15:00 Wednesday 17:00 - 19:00 Thursday	L.	WEEKLY TUTORING SCHEDULE	11:00 - 12:00 Monday 16:00 - 17:00 Thursday

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.

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 / Hours / Evaluation Instruments

CURRICULAR UNITS	CONTENTS	TEACHING HOURS	HOURS OF INTERNSHIP AND EXPERIMENTAL LEARNING	HOURS OF INDEPENDENT LEARNING	EVALUATION INSTRUMENTS
Unit 1. Ordinary differential equations in physics	Ordinary differential equations, and initial value problems	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Euler methods, Runge-Kutta methods, and applications	4	4	4	Classwork (quizzes), homework (assignments), and exams.
	Boundary value problems, shooting and finite difference methods, applications	4	4	4	Classwork (quizzes), homework (assignments), and exams.
Unit 2. Software design using object-oriented programming	Python classes and modules	2	2	2	Classwork (quizzes), homework (assignments), and exams.
	Standalone modules and python packaging for physics	4	4	4	Classwork (quizzes), homework (assignments), and exams.
	Timing tests and efficient coding	2	2	2	Classwork (quizzes), homework (assignments), and exams.
Unit 3. Parallel computing for physics	High-performance computing (HPC) and job managers	3	3	3	Classwork (quizzes), homework (assignments), and exams.

	Parallel computing, CPU/GPU, multiprocessing and joblib	4	4	4	Classwork (quizzes), homework (assignments), and exams.
	The Message Passage Interface (MPI) and Application Programming Interfaces (APIs)	4	4	4	Classwork (quizzes), homework (assignments), and exams.
Unit 4. Partial differential equations in physics	Partial differential equations, generalities, classification and solving methods	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Parabolic and elliptical problems. Heat and Poisson equations	4	4	4	Classwork (quizzes), homework (assignments), and exams.
	Hyperbolic problems: Introduction to computational fluid dynamics (CFD)	2	2	2	Classwork (quizzes), homework (assignments), and exams.
Unit 5. Special topics in computational physics	Advection, thermodynamic simulations and introduction to molecular dynamics	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Nonlinear dynamics, chaotic systems, fractals and statistical growth	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Introduction to machine learning	3	3	3	Classwork (quizzes), homework (assignments), and exams.
TOTAL		48	48	48	144

7. Learning outcomes of the course

LEARNING OUTCOMES		STUDENT IS REQUIRED TO: (EVIDENCE OF LEARNING)
A.	Improve object-oriented programming skills to solve physics problems within Linux environments.	Submit quizzes on reading material and code developed in class (classwork). Hand in routines of code, scripts, and group reports (homework). Solve programming problems in exams (mid-term and final exams).

B.	Design algorithms, implement and debug parallel software within the message passing interface in Python language.	Submit quizzes on reading material and code developed in class (classwork). Hand in routines of code, scripts, and group reports (homework). Solve programming problems in exams (mid-term and final exams).
C.	Apply numerical methods and computational techniques to simulate physical systems via reusable and extensible code packages.	Submit quizzes on reading material and code developed in class (classwork). Hand in routines of code, scripts, and group reports (homework). Solve programming problems in exams (mid-term and final exams).
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.	Submit quizzes on reading material and code developed in class (classwork). Hand in routines of code, scripts, and group reports (homework). Solve programming problems in exams (mid-term and final exams).

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)**		**Total**
1 Midterm Exam	30 %	Homework average (code routines and project reports)	20 %	Classwork average (reading quizzes and programming exercises)	20 %	1 Final Exam	30 %	100 %
Subtotal	30 %	Subtotal	20 %	Subtotal	20 %	Subtotal	30 %	
10.2. Makeup Exam								
N/A								

11. General considerations

- Students are responsible for ensuring the academic integrity of their submitted assignments and exams.
- Cheating in exams, plagiarising, and copying solutions from other students, from solution manuals, or from previous years' solutions are all breaches of academic integrity.
- Academic misconduct will be penalised according to the University's regulations.
- Assignment deadlines and exam dates will be discussed and agreed upon in class. Once fixed, they are hard deadlines.

(*) The teaching staff will register the scores generated up to mid-semester in Moodle, by the deadline set in the academic calendar for this term. When defining the weights of each item, it is necessary to observe what is established in Article 35 of the Internal Rules of the Academic Regime of UITEY.

(**) For courses without a laboratory component, indicate: N/A.

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