YACHAY TECH UNIVERSITY

SYLLABUS

1. General Information					
A.	**SCHOOL**	Physical Sciences and Nanotechnology	**B.**	**MAJOR**	Physics
C.	**COURSE**	Computational Physics I	**D.**	**CODE**	ECFN1015
E.	**LEVEL**	7th	**F.**	**ACADEMIC TERM**	Second Semester 2024
G.	**CURRICULAR UNIT**	Professional	**H.**	**STUDY MODE**	In person
I.	**TOTAL HOURS**	176	**J.**	**PROFESSORS**	Wladimir Eduardo Banda Barragán
K.	**WEEKLY CLASS SCHEDULE**	09:00 - 11:00 Monday 07:00 - 09:00 Tuesday 14:00 - 16:00 Friday	**L.**	**WEEKLY TUTORING SCHEDULE**	14:00 - 15:00 Tuesday 15:00 - 16:00 Wednesday

2. Prerequisites and Corequisites			
PREREQUISITES		COREQUISITES	
COURSE	**Code**	**COURSE**	**Code**
Numerical Methods	ECMC1006		
Introduction to Programming	ECMC1008		

3. Course Description

This course provides an introduction to basic methods and techniques used in computational physics as well as an overview of recent progress made in several areas of scientific computing. The course describes basic concepts of object-oriented programming and includes detailed step-by-step examples of how to optimally utilise computers and programming languages to solve problems in physics. Topics range from data analysis, approximation and optimisation of functions, through numerical calculus and differential equations, to matrix operations and spectral analysis. Each section of the course includes practical examples on different areas of science and technology in which computational physics has played a major role in the last decade.

4. Course Contribution to professional training

This course helps students to build programming skills for the design and implementation of software dedicated to applications in physical sciences.

5. Course objectives

- Develop object-oriented programming skills for scientific computing within Linux environments.
- Design algorithms and implement software (in Python) dedicated to data analysis and visualisation.
- Apply numerical methods and computational techniques to model physical systems, analyse images, and carry out Fourier analysis.
- Use computational methods for research applications on 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. Data analysis and visualisation	Introduction to computer science and scientific programming	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Errors and uncertainties in computations, computer algorithms, and languages	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Object-oriented programming, data input/output, plotting, statistics, data fitting, and regression	4	4	2	Classwork (quizzes), homework (assignments), and exams.
	Function approximation, interpolation and extrapolation, Spline approximation.	4	4	2	Classwork (quizzes), homework (assignments), and exams.
Unit 2. Linear algebra and matrices in physics	Array programming, vectors, matrices, and images	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Matrix operations, basic image processing, and visualisation tools	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Linear equation systems and eigenvalue problems	3	3	2	Classwork (quizzes), homework (assignments), and exams.

	Iterative methods for linear and non-linear systems	3	3	2	Classwork (quizzes), homework (assignments), and exams.
Unit 3. Numerical calculus	Numerical differentiation	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Numerical integration	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Derivatives and integrals on gridded data	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Numerical optimisation, root-finding and extreme values of functions.	3	3	3	Classwork (quizzes), homework (assignments), and exams.
Unit 4. Spectral analysis	Fourier analysis	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Discrete Fourier transform and the Fast Fourier transform algorithm	3	3	2	Classwork (quizzes), homework (assignments), and exams.
	Fourier filtering and denoising.	4	4	2	Classwork (quizzes), homework (assignments), and exams.
	Wavelet analysis and discrete wavelet transform	4	4	2	Classwork (quizzes), homework (assignments), and exams.
Unit 5. Monte Carlo techniques and differential equations	Introduction to Monte Carlo methods	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Monte Carlo simulations	3	3	3	Classwork (quizzes), homework (assignments), and exams.

	Monte Carlo applications, random walks and radioactive decay	3	3	3	Classwork (quizzes), homework (assignments), and exams.
	Introduction to ordinary differential equations and initial-value problems	3	3	3	Classwork (quizzes), homework (assignments), and exams.
TOTAL		64	64	48	176

7. Learning outcomes of the course

LEARNING OUTCOMES		STUDENT IS REQUIRED TO: (EVIDENCE OF LEARNING)
A.	Develop object-oriented programming skills for scientific computing 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 and implement software (in Python) dedicated to data analysis and visualisation.	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 model physical systems, analyse images, and carry out Fourier analysis.	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 computational methods for research applications on 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: Firmado electrónicamente por: WLADIMIR EDUARDO BANDA BARRAGAN	SIGNATURE AND DATE: CLARA INES ROJAS CELY Digitally signed by CLARA INES ROJAS CELY Date: 2024.09.17 07:53:40 -05'00'	SIGNATURE AND DATE: GEMA GONZALEZ VAZQUEZ Firmado digitalmente por GEMA GONZALEZ VAZQUEZ Fecha: 2024.09.17 08:44:51 -05'00'