

**LEHMAN COLLEGE
OF THE
CITY UNIVERSITY OF NEW YORK**

DEPARTMENT OF PHYSICS AND ASTRONOMY

CURRICULUM CHANGE

1. **Type of change:** Experimental Course

2.

Department(s)	Physics & Astronomy
Career	☒ Undergraduate ☐ Graduate
Academic Level	☒ Regular ☐ Compensatory ☐ Developmental ☐ Remedial
Subject Area	Scientific Computing Methods
Course Prefix & Number	PHY 248
Course Title	Introductory Scientific Computing
Description	Fundamentals of scientific computing with Julia or Python. Numerical solution of algebraic and ordinary differential equations, numerical minimization, numerical integration, Monte Carlo methods, N-body simulations. Algorithm efficiency and parallelization. Examples in physics, chemistry and biology.
Pre/ Co Requisites	PREREQ: Calculus I (MAT175) and Calculus II (MAT176)
Credits	3
Hours	3
Liberal Arts	☐ Yes ☒ No
Course Attribute (e.g. Writing Intensive, WAC, etc)	
General Education Component	☒ Not Applicable ☐ Required ☐ English Composition ☐ Mathematics ☐ Science ☐ Flexible ☐ World Cultures ☐ US Experience in its Diversity ☐ Creative Expression ☐ Individual and Society ☐ Scientific World

3. Rationale:

While STEM fields were born in a time without computers, it is now hard to imagine making progress without them. Many problems across disciplines benefit from numerical analysis and simulations. Currently, no course in the department of Physics & Astronomy or other natural sciences serves as an introduction to the methods of scientific computing. This course addresses that gap. The low entry level and diversity in examples across disciplines will allow students across many departments to gain familiarity with the power of computational tools in their fields, strengthening their preparation for a future within or outside of their current fields.

4. Learning Outcomes (By the end of the course students will be expected to):

- Set up a local Julia/Python environment, run scripts, and access their output.
- Know the basic elements of programming for scientific computing: variables, arrays, loops and functions.
- Be familiar with the physical constraints for scientific computing (speed/time, memory and storage), and estimate the resources needed for a program.
- Have familiarity with the kind of problems that can be tackled with computers.
- Implement their own algorithms to solve simple mathematical problems.
- Use existing algorithms to tackle complex problems.

5. Date of Departmental Approval: September 1, 2026