Course Title: Computational Methods (using Python)
Course Lecturer: Mr. Mayhew Steyn
Course credits: 2
Lecturer contact hours: 40
Tutorial/practical hours: 20

1) Course overview:
In this course we will cover the basics of Python, a few popular packages relevant to astronomy and physics, and general numerical methods used in science. The focus of this course is on the practical application of this programming knowledge, and most of the contact time will be dedicated to working through exercise and tutorials.

Each lecture will consist of a pre-reading component from the course notes, an in-class revision of this, and time to work on exercises related to the contents of the readings (not assessed).

Assessments will consist of tutorials to be completed in class-time, and take-home write-ups reflecting on these. In these tutorials you will be given a problem to work through individually by using Python programming. These tutorials will make use of the materials covered in the lectures, exercises and previous tutorials.

2) Course breakdown/syllabus:
- Use of the Python Standard Library:
  - Variables, operators, data types and data structures.
  - Control flow: If Statements and Loops
  - Functions
  - File input/output
- Scientific Python Packages:
  - Numpy (arrays, special functions, and random numbers)
  - Matplotlib (plotting)
  - Scipy (curve fitting, numerical algorithms)
  - Astropy (astronomy focused, dimensional analysis, image processing, etc)
- Numerical Methods:
  - Linear regression and curve fitting with Scipy
  - Solving ordinary-differential equations (Euler and Runge-Kutta methods)
  - Integration (midpoint, trapezoidal and Simpson`s rules)
  - Root finding (bisection, secant and Newton-Raphson methods)

3) Resources:
Course notes (website and PDF formats) and all other resources provided.

4) Breakdown of practicals/tutorials:
In lectures you will be given exercises to work on with the guidance of the lecturer and tutors.
In the tutorial sessions you will be given a tutorial to work through with guidance from tutors.
In addition, you will be tasked with writing reflections on the tutorials.

5) Additional skills to be developed during the course:
Python programming (writing original code), scientific writing/reporting (using LaTeX), numerical problem solving (transforming physics into code).

6) Assessment:

- Tutorials: 75%
- Written reflections: 25%