

# Design Document

Computer Modelling 2016-17

Due: 5pm Thursday, Week 3, Semester 2

## 1 Aims

Before you start implementing your project, you should complete a *design document* that outlines your plans of implementation. As such, the design document should make clear the proposed structure of your project program to anyone who is not involved in the programming, but has a grasp on the Python language and general concepts of programming.

## 2 Task

Write a document that outlines how you propose to implement the project tasks. That includes, but is not necessarily limited to:

- An overview of what your program will be able to do.
- The class structure of your program. Which non-standard, user-written classes will be used, and how are they connected?
- Descriptions of said classes: their properties and the methods that will be (or already are) implemented.
- Description of the main code, which outlines how your program will fulfil the tasks required.
- Descriptions of all functions that will be implemented to support the main code.

## 3 Detailed Instructions

Start by reading carefully the appropriate project brief, to learn what *functionality* you need to implement in a code that fulfills all project tasks.

Then, develop a *structure* for the project code: will you re-use classes you already wrote, will you need more dedicated classes, and how will those be laid out? How will

the main program be structured, and which computations will be in-line or done in dedicated functions?

Next, think about how you will implement the requested functionality. Either project asks you to perform specific  $N$ -body simulations. These simulations need well-specified initial conditions – how will those be set up and communicated to the code; need to run the simulations – what is the algorithmic procedure to do that; and retrieve data, or observables from the simulations – how will those be obtained.

The design document should include your decisions regarding the points above. For pre-written or yet-to-be-written classes, the properties and functionality of methods must be laid out clearly; think about the requirements for a well-written `pydoc` documentation. For the main code, you should describe any non-trivial algorithms, such that an external person has enough information to decide whether or not your planned implementation will work. Feedback by markers on this last point could save you a lot of time and trouble, so make sure your descriptions are detailed enough to allow for productive feedback.

For your guidance, there are two example design documents on the course LEARN page that describe the same (fictitious) project. One is better than the other.

## 4 Submission

Submit your design document (preferably in PDF format) through the Turnitin link on the course LEARN page, by **5pm on Thursday, week 3 of semester 2**. Only one submission per group is necessary.

## 5 Marking Scheme

This assignment counts for 10% of your total course mark.

1. Aim of the project [2]
2. Description of classes [6]
  - Overall structure
  - Descriptions of properties and methods
3. Description of main program [9]
  - Functionality
  - Algorithms and methods to be implemented
4. Layout, language and style [3]

Total: 20 marks.