# Review: Section 2/Exam 2:

## Use of fsolve

* 1. structure: fsolve(callback, initial guess, (args))
  2. callback(Q, \*args)
     1. Q: array of values/unknowns that fsolve is changing to make equations inside the callback function equal zero.
     2. \*args: allows the callback to accept an unlimited number of positional arguments.
     3. \*\*kwargs: allows a function to accept an unlimited number of keyword arguments.
     4. return: a list of the values of the functions defined in the callback.
  3. initial guess: array of initial values (guesses); equal in length to Q.
  4. (args): a tuple of values that gets unpacked and passed to callback as positional arguments; hence \*args in the callback.

## Use of solve\_ivp

1. structure: solve\_ivp(callback, [start time, end time], initial conditions, t\_eval, (args))
2. callback(t, X, \*args)
   * 1. X: array of state variables
     2. t: the time specific to this instance of the function.
     3. \*args: allows a function to accept an unlimited number of positional arguments.
     4. return: the derivatives of the state variables.
3. initial conditions: array of initial values for the state variables.
4. (args): a tuple of values that gets unpacked and passed to callback as positional arguments; hence \*args in the callback.

# Section 3/Exam 2

## Reading data from a table with numpy.loadtxt

1. skiprows
2. unpack

## Interpolating data

1. 1-D interpolation with scipy.interpolate.interp1d
2. Interpolate on grid data with scipy.interpolate.griddata

## Object oriented programming (OOP)

1. A class a general description of ‘something’ we want to model (e.g., a bolt)
2. An object a particular instance of a class.
3. classes contain: 1. a constructor, 2. properties, 3. methods
4. We can have many instances (objects) of a class in a program.
5. inheritance: classes can inherit properties and methods from parents.
6. polymorphism: methods of the parent class can be made to behave differently in the child.
7. classes can have objects of other classes as properties

# Exam 2: week of 11 March to 18 March

1. Modules, numpy, scipy, matplotlib, numerical integration
2. fsolve, solve\_ivp
3. reading and interpolating data, object oriented programming