**Worksheet Open ‘Lec09 student copy’** (use with set text, Negnevitsky 3rd ed. 218)

* **Slide 11 Basic genetic algorithms:** Remind yourself of the terms; represent problem domain, encode a chromosome of a fixed length, chose the size of the of the chromosome population N, the crossover probability pc, mutation probability pm. Define a fitness function for individual chromosome to select for reproduction. Randomly generate an initial population of chromosomes. Calculate the fitness of each individual chromosome. Select a pair of chromosomes for mating from population. Parent chromosomes are selected with a probability related to fitness. Create pair of offspring chromosomes using genetic operators crossover and mutation, placing in new population replacing some of the original members. Test the fitness of the new generation. Continue to evolve the population towards solution.

p.222

* **Exam question 09-1:** With a particular application in mind explain why you would chose particular values for particular parameters and if any of them are more likely to be varied than others.
* **Slide 17 The fitness function and chromosome locations of the (15x − x2) case study:** Explain why it is not optimum solution.

p.225

* **MATLAB:** GA\_1
* **Exam question 09-2:** MATLABWhat is the problem and how is it being encoded?

p.224

Slide 34 Performance graphs for 100 generations of 6 chromosomes: What does this graph represent?

p.230

* **MATLAB:** GA\_2
* **Exam question 09-3**: MATLABWhat is the problem and how is it being encoded?
* **Slides 37-40 Case study - maintenance scheduling:** In groups of three, look at slides 37 to 40 but do not look any further until you have considered how to encode the genes and what they represent in terms of the schedule and how they will reside on the chromosome, it may take some time. Only when you have a sense of how to encode the problem move on to the offered solution.

p.235

• **Exam question 09-4**: Discuss the merits of the following statements: If you have a fitness function you can evolve anything; encoding the problem is where most of the work resides in a genetic algorithm.

* **Demo task 29** GA\_1 p.222
* **Wiki 29** Is GA\_1 a sensible application? P.222
* **Demo task 30** GA\_2 p.227
* **Wiki 30** Create guidelines for configuring GA\_2 for the best performance? p.227