

What can you tell me about the 2015 BIC exam ?

Nothing. However, past papers are available - see the end of the page and associated notes at:

<http://www.macs.hw.ac.uk/~dwcorne/Teaching/bic.html>

It is fair to point out a few things though, which could be gleaned by examining past papers:

- The BIC exam has always been “Answer 3 questions”, from a set of 4 questions
- You can generally expect each lecturer to have at least one question;

DWC’s part of the Bio-Inspired Computing module: 2015: What should I know?

To be read in conjunction with the table at <http://www.macs.hw.ac.uk/~dwcorne/Teaching/bic.html>

You should read/know everything in the slides and in the ‘mandatory’ extra bits.

In a little more detail, perhaps the following helps. You should know ...

- What classical computing is good at
 - What classical computing is bad at
 - What biological systems are good at
 - ... hence the motivation for bio-inspired computing
 - What pattern recognition / classification means
 - What optimisation means
-
- You should be able to define all of these :
 - Optimisation,
 - Classification
 - Fitness function
 - Exact algorithm
 - Approximate algorithm
 - Problem complexity
 - Easy problem
 - Hard problem
 - Polynomial complexity
 - Exponential complexity
 - Minimal spanning tree

- Prim's algorithm
 - Typical complexity of real-world problems
 - Typical performance of approximate methods
-
- the basic ingredients of any EA (i.e. a 'generic' EA)
 - about the varieties of different EAs – i.e. in what aspects they tend to differ from each other
 - you should be able to indicate several different types of application of EAs
 - you should be able to write basic pseudocode for various types of EA, that use the elements and variations that are described in the lecture slides (including mandatory additional). Including steady-state, generational, elitist, different types of selection, and so on.
 - You should be able to write the pseudocode for hill-climbing;
 - what are selection, variation and replacement in the context of EAs
 - what encoding means (same as 'representation')
 - simple and typical encodings for certain problems – e.g. bin-packing, travelling salesperson problem
 - what is a 'landscape', in this context
 - what is a 'neighbourhood' in this context
 - what is local search
 - pseudocode for local search
 - basic types of local search
 - what is a direct encoding, and what is an indirect encoding
 - what is a constructive algorithm
 - features of direct vs indirect encodings
 - how you might develop an indirect encoding using a constructive algorithm
 - names and basic shapes of particular landscape features
 - everything in the encoding, operators and selection ppts. i.e. all of the specific encodings selection methods and operators described therein
 - The basic idea of swarm intelligence
 - What 'emergent' means in this context
 - Simple swarm-based rules for clustering
 - The three main things that computer science gets from swarm intelligence
 - Reynolds rules – the rules themselves, the perceptual system, and what they are used for
 - What is stigmergy
 - How ants seem to find the shortest path between nest and food source
 - The basic ACO algorithm, how it is applied, its main parameters and their effects
 - The details: the transition and update rules, and what tends to be added to ACO to get the best results.

- The basic pseudocode for a PSO algorithm
- The different types of neighbourhood for PSO (note that neighbourhood means something different in this context)
- The main parameters in PSO, what they relate to, and their effects on the search

Note that you should know the above things well enough, where relevant, to be able to apply them. In the context of the examination, this means, for example, that you should be able to:

- Describe how you might apply an EA (or PSO, or ACO, or cellular automata) to a specific task I describe in an exam question.
- Work out the sizes of neighbourhoods in particular specified cases (e.g. the exam question might describe an encoding and an operator, and ask you to calculate the size of the neighbourhood, or it may ask for a related calculation)
- Show me how a particular constructive algorithm works step by step
- Invent your own constructive algorithm for a particular task I describe
- ... or anything else I might think up which would test your knowledge of the above.

A note about DWC Mandatory additional material

The mandatory additional material consists of three sets of ppt slides:

- [Hard and Easy optimization problems](#),
- more encodings [ppt](#),
- Selection [ppt](#) ,
- Operators [ppt](#)

And one paper: <http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>

The mandatory ppt material has been incorporated in the 'What should I know' section above. The paper is for you to read, enjoy and digest, so that you understand more about the background and application of Reynold's rules. If there is ever an exam question that calls on your knowledge of the paper, it will not depend on any specific details, but it will be based on you being able to do more than simply restate what was in the 'Reynolds' rules' lecture. E.g. " Briefly discuss one of the following three aspects of Reynolds' rules: simulated perception; obstacle avoidance; differences between natural and simulated flocks."