COMP102: Computational Thinking

15 credits

Firstly, you will be introduced to the concept of computational thinking. In particular, you will consider the question 'What is it to compute?'. A number of key concepts are defined – such as computational problem, algorithm and abstraction – and their application is illustrated. You will be given examples of computational problems and their solutions in a wide variety of fields, ranging from physics and biology to economics and sports. You will also start using the Python programming language.

Secondly, you will be introduced to tools and techniques for:

- creating abstractions that represent a problem
- devising algorithms that solve a problem efficiently.

A range of standard data structures and algorithms for sorting, searching and optimisation will be covered and illustrated with practical examples. You will also encounter notions such as Big-O notation, induction and recursion.

Finally, you will further develop your understanding of sets, functions, logic and proofs. In particular, some of the ideas – that you will have been introduced to informally – will now be presented using formal mathematical notation. This will be in the context of concrete applications, such as databases. At this point, you'll also learn about the limitations of computational problem solving (non-computability and computational complexity) and recent developments in computing, such as quantum computing.

Skills Contributions

- DTAN
- PROG

Origin

based on Open University Module M269: Algorithms, Data Structures and Computability.