Computing Extended Reading List

Algorithms to Live By written by Tom Griffiths and Brian Christian (100)

A fascinating exploration of how insights from computer algorithms can be applied to our everyday lives, helping to solve common decision-making problems and illuminate the workings of the human mind.

Code the Hidden Language of Computer Hardware and Software (004.23)

What do flashlights, the British invasion, black cats, and seesaws have to do with computers? In CODE, they show us the ingenious ways we manipulate language and invent new means of communicating with each other. And through CODE, we see how this ingenuity and our very human compulsion to communicate have driven the technological innovations of the past two centuries. Using everyday objects and familiar language systems such as Braille and Morse code, author Charles Petzold weaves an illuminating narrative for anyone who's ever wondered about the secret inner life of computers and other smart machines.

Computing by Geoffrey Knott and Nick Waites (004)

"I have used this book for my A-Level studies and have found it useful and has helped my understanding of the subject. From hardware to software, binary to complex applications, this book has helped me and I would highly recommend it to those interested in computing or thinking about taking computing on as a subject. I shall, without doubt, refer to this book at University."

Computing projects in Visual Basic.net by Derek Christopher (005.133)

Written mainly for students of A Level computing

Introductory Computer Science by A K Dewdney (004)

This introductory text provides both a foundation in a popular programming language (Turbo Pascal) and an introduction to the principles and applications of the field. In addition to providing an overview of computer science via programming, this book stresses applications that demonstrate computers' many roles in our lives and concepts that shape the design of new software and hardware.

Learning to Program in Pascal and Delphi by Sylvia Langfield (005.133)

Written for students on advanced level computing courses who require a knowledge of programming.

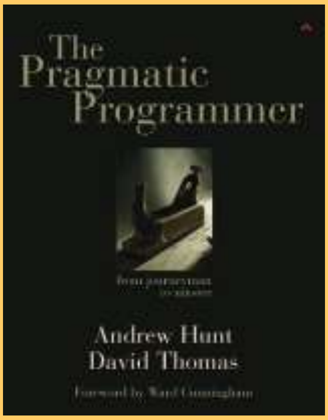
Life 3.0: being human in the age of artificial intelligence by Max Tegmark (006.3)

Will superhuman intelligence be our slave, or become our god? Taking us to the heart of the latest thinking about AI, Max Tegmark, the MIT professor whose work has helped mainstream research on how to keep AI beneficial, separates myths from reality, utopias from dystopias, to explore the next phase of our existence. How can we grow our prosperity through automation, without leaving people lacking income or purpose? How can we ensure that future AI systems do what we want without crashing, malfunctioning or getting hacked?

Novacene: The Coming of Age of Hyperintelligence by James Lovelock (570.1)

James Lovelock, creator of the Gaia hypothesis and the greatest environmental thinker of our time, has produced an astounding new theory about future of life on Earth. He argues that the anthropocene - the age in which humans acquired planetary-scale technologies - is, after 300 years, coming to an end. A new age - the novacene - has already begun. New beings will emerge from existing artificial intelligence systems.

The Pragmatic Programmer by David Thomas and Andrew Hunt (005.1)



The Pragmatic Programmer cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, The Pragmatic Programmer illustrates the best practices and major pitfalls of many different aspects of software development.