RESEARCH　　　　　　　　　　　　　　　　　　　　　　　　　　**Open Access**

Compulsory coding in education: liberal-humanism, Baudrillard and the 'problem' of abstraction

Laurence Tamatea

Correspondence: Laurence.
tamatea@cdu.edu.au
College of Education, Charles
Darwin University, Darwin, Australia

Abstract

Internationally, coding is increasingly introduced into primary and junior high schools (children generally aged between 5 and 15) on a compulsory basis, though not all stakeholders support this 'initiative'. In response to the public reception, discussion highlights popular argument around compulsory coding in school education. This is an argument between those supportive (hereafter referred to as the Yes case) and those unsupportive of compulsory coding (hereafter referred to as the No case). But more than simply produce a list of arguments, this discussion contributes to our understanding of this reception by identifying the 'discourses' deployed by both cases (namely, digital ubiquity, disadvantage, and habits of mind discourses) and by providing theoretical framings through which these discourses and their potential implications might be differently understood. Using critical discourse analysis to unpack these discourses shows that while both cases hold to key tenets of liberal-humanism, a commitment to the individual subject, liberty and full participation in the social, it is the Yes case with its stronger commitment to children engaging in abstraction that seems to challenge these. Discussion of this difference is framed by the work of Baudrillard around abstraction, not to 'prove' the validity of Baudrillard's thesis concerning the consequences of humanity's deepening engagement with abstraction, but to provide a broader understanding of this debate, in relation to a trajectory of engagement with abstraction that seems set to intensify.

Introduction

Digital code underwrites war machinery (Howard, 2013), the stock market (Clarke, 2013), autonomous stealth drones (Northrop Grumman, 2015), robotics (Boston Dynamics, 2017) and artificial intelligence (Evlin, 2017). And while not less than two decades ago we proclaimed AI could not 'really' outperform humans, we now speak of the need to 'check' its rise (Rawlinson, 2015; Vincent, 2017a). Our world is inscribed by the digital, and education has responded by introducing compulsory coding in schools (Chambers, 2014; Estrin, 2017; Euractive, 2015; Hamilton-Smith, 2017; Smith, 2016; Turula, 2017; Verma, 2016; Woolf, 2014). This response, however, has not been without debate, within which some argue the significance of coding to self and society, while others, its irrelevance. Although school education is now often the subject of debate, the significance of this debate emerges in the extent to which the politico-philosophical underpinnings of the modern education

project, namely liberal-humanism, are simultaneously championed and yet (possibly) undermined. Exploring this seeming contradiction, discussion highlights argument for and against compulsory coding in school education, represented here as the Yes and No cases. But more than simply producing a list of arguments, discussion identifies the dominant discourses deployed in this debate.

More than words, discourse comprises the strategic deployment of socially shared patterns of words in constructing claims to truth (Foucault, 1981). And in the debate around compulsory coding in schools, the dominant discourses comprise those around digital ubiquities, disadvantage and habits of mind. More specifically, Yes case discourse is inscribed by positive liberty, and the No case, negative liberty. The Yes case maintains that the realization of the individual and their freedom emerges from compulsory exposure to state-sanctioned knowledge, while the No case holds that these only emerge where the individual self-selects valued knowledge.

Despite this difference, both cases reproduce key tenets of liberal-humanism, namely a commitment to the individual subject, liberty and participation in the social. But this consensus breaks down with the Yes case's deployment of the habits of mind discourse, wherein prioritization of computational thinking constructs the individual as less analogue, independent and inscribed by liberty than liberal-humanism perhaps ever imagined. With this, the Yes case is inscribed by a contradictory 'logic', such that I ask: How can the Yes case deploy key tenets of liberal-humanism to advocate compulsory coding in schools, when the very thing argued for—deeper engagement with the digital and consequently abstraction—potentially erodes such tenets? This question, in part, motivates Kissinger's discussion of the Enlightenment and the rise of artificial intelligence (2018), wherein he reflects upon how the Age of Reason's penchant for scientific knowledge not only displaced the prior 'liturgical order', but now threatens to undermine the very project from which it was birthed.

Kissinger concludes, noting that while 'the Enlightenment started with essentially philosophical insights spread by a new technology, our [current] period is moving in the opposite direction, [generating] a potentially dominating technology in search of a guiding philosophy' (2018 para 29). It is this contradiction which is present in the debate around compulsory coding in schools, particularly so in the Yes case, such that deeper analysis of this debate requires a robust theoretical if not philosophical framing, taking our understanding of the positions and protagonists beyond seeing such as simply Luddite-like or progressive. To this end, the discussion draws upon Baudrillard (1983; 1993; 1995; 2009) to reflect upon the possible implications of the quantitative 'logic' of deepening abstraction, with which the Yes case's desire for children to increasingly 'interface' with computers seems to align.

Discussion concludes that while the 'quantitative' has always informed liberal-humanism, such that AI could emerge from the Age of Reason (Kissinger, 2018, para 6), the strengthening alignment with the logic of abstraction signalled in Yes case' habits of mind discourse, might otherwise be considered—drawing upon Baudrillard—to be synonymous with humanity's move into times largely inconsistent with the Enlightenment and indeed liberal-humanism's 'traditional' construction of humanity. Baudrillard is used here, not to establish what this new philosophy of humankind might be, a project beyond the scope of this paper. Rather, Baudrillard is used as an alternative framing for understanding the debate around compulsory coding in schools, allowing consideration of how the coding initiative

challenges the existing dominant philosophy, namely liberal-humanism, upon which both it and indeed the modern education project rests.

Literature: the popular voice—largely ignored

Coding in schools is not a new phenomenon, being a curriculum area introduced as early as the 1970s in some settings (Knobelsdorf et al., 2015) and during the 1980s and 1990s in others (Cartelli, 2002; Popat & Starkey, 2019). While these instances have been implemented with varying success over the past 30 to 40 years, the most recent 'initiative' arguably gained momentum from the first decade of the twenty-first century (Bresnihan, Millwood, Oldham, Strong, & Wilson, 2015), as part of a larger coding for all movement (Dimeo, 2017; Shein, 2014; Thoreau, 2017). Whereas the coding agenda was once more so associated with programmers and computer societies seeking discipline exposure (Jovanov, Stankov, Mihova, Ristov, & Gusev, 2016), it is now an initiative shared more broadly by states, industry, entrepreneurs and presidents (Smith, 2016). Related to this has been the emergence of the creative industries economy (Gouvea & Vora, 2018; Tamatea & Paramatasari, 2018) and arguably a populist change in the discursive representation of the programmer (Kainulainen, 2013; Pritchard, 2016). Now it seems everyone can or should code. Code bootcamps are now 'big business' (Clark, 2018), while online coding courses are ubiquitous.

Coding has come 'out of the lab' and is now to be engaged by the 'masses', who, as I show below, have an opinion about this. Less apparent, however, is the voice of the academy regarding this popular mass response to 'coding for all' in schools. Certainly, there has been formal research around coding and education. Studies explore how to teach programming in schools (Mathrani & Agate, 2016; Pellas & Efstratios, 2017), universities (Abdul-Rahman & Bendeict, 2014) and now in early childhood settings (Campbell & Walsh, 2017). Studies acknowledge the difficulty of learning coding (Cardell-Oliver, 2014, Bain and Barnes 2014; Guzdial, Hohmann, Konneman, Walton, & Soloway, 1998) and, related to this, how coding might be best introduced in education settings, in terms of how to best teach coding (Merkouris, Chorianopoulo, & Kameas, 2017; Hiltunen, 2016) and which languages and technologies are best introduced first (Pellas & Efstratios, 2017). Much of this pedagogy-focused research (Hutchison, Nadolny, & Estapa, 2015) suggests students are best introduced to coding not through text-based languages such as C++, C# or Java, but through visual and block-based languages like Scratch (Kalelioğlu & Yasemen, 2014; Saez-Lopez, Gonzalez, & Vazquez-Cano, 2016; Van Zyl, Elsa Mentz, & Havenga, 2016). The emergence of visual coding environments has in part been responsible for the traction of the coding for all initiative (Popat & Starkey, 2019, p. 365; Wong, Cheung, Ching, & Huen, 2015).

While research-informed discussion focused specifically on reasons why coding is now more prominent in the school curriculum is arguably still limited; notwithstanding policy-linked reports invariably constructing an economic case (Deloitte Access Economics, 2016), these reasons have occasionally been articulated as part of a broader discussion of coding in schools, particularly where coding is a compulsory subject. This literature shows that while some reasons have changed over time, others have remained largely the same (Knobelsdorf et al., 2015; Jovanov et al., Hubwieser, 2012). In their discussion of coding in German school curriculum from the 1970s, Knobelsdorf et al. note that:

> Witnessing the increasing role of Information Technology (IT), stakeholders began arguing for the inclusion of CS in secondary education. On the bases of economic growth, societal welfare, and the stability of the labor market, they argued that Germany would need computer scientists and employees to fill IT positions across sec- tors. At a fundamental level, being able to solve problems was viewed as an important skill that all students should possess, even if at a basic level (2016 p. 5).

These reasons are not too dissimilar from those informing the current initiative.

Similarly, the introduction of 'coding' in Italian state schools can be attributed to a 1980 report (*Inquiry on Computer Science*) identifying a need for more employees with information technology skills (Cartelli, 2002). But beyond employment, the report proposed that learning to code would result in a number of learning outcomes in domains not related to coding (p. 36). These outcomes are today referred to as spill-over or transfer skills, such that Popat and Starkey (2019) ask: Are students learning coding or coding to learn? Answering their question, they identify six educational outcomes including problem-solving through mathematical concepts, social skills including collaboration, self-management/active learning, critical thinking and 'Academic skills (NOT including mathematical or computer science/ programming related skills)' (p. 367).

Central to this question is also the notion of (digital/information) 'literacy', where learning to code results in acquisition of a range of skills as significant to the learner's capacity to participate in the twenty-first century society as acquisition of 'traditional' literacy was to full social participation in earlier times (Bresnihan et al., 2015; Hutchison et al., 2015; Popat & Starkey, 2019; Vee, 2013). Key among these transfer skills is 'computational thinking' (Wing, 2010), and to a lesser degree 'computational participation' (Kotsopoulos et al., 2017; Burke, O'Bryne & Kafai, 2016; Fields, Giang, & Kafai, 2014; Kafai & Burke, 2014; Kafai & Burke, 2013). But whereas computational participation is ground Understanding computer programming as a literacyed in the claim that coding is collaborative providing students with the skills to participate in a networked society (Popat & Starkey, 2019, p. 368), the notion of computational thinking makes grander claims in relation to transfer skills.

Notwithstanding disagreement around what computational thinking more precisely comprises (Shute, Sun, & Asbell-Clarke, 2017), it is a framework for understanding (if not constructing) human-computing machine relations well discussed within formal literature (Gover & Pea, 2013). According to Shute et al. (2017), the notion emerged from 'the constructionist work of Seymour Papert (1980, 1991), being first coined as a term in a seminal article by Wing (2006)' (p. 143). Consequently, Wing's work now is often associated with computational thinking (Wing, 2008, 2010, 2016). Broadly, computational thinking is grounded in a number of elements associated with cognitive processes, one specifically related to abstraction—the ability to 'model the core aspects of complex problems or systems' (Shute et al., 2017, p. 145; Wing, 2010). Indeed, Wing advocates the value of computational thinking to life more generally, and utility in disciplines beyond science, maths and engineering, such that it is considered a new 'literacy' (Wing, 2010, 2016).

Those advocating compulsory coding often draw parallels between present-day programmers and ancient world Scribes, arguing that just as literacy is now a skill possessed by most as opposed to the few, so should computational thinking/literacy be

extended to most (Raja, 2014; Shute et al., 2017; Vee, 2013). Crow (2014) argues that 'not knowing the language of computers will be as challenging as being illiterate or innumerate are today' (p. 2). Thus, computational thinking is regarded as a 'universal competence, which should be added to every child's analytical ability as a vital ingredient of their school learning' (Voogt, Fisser, Good, Mishra, & Yadav, 2015, p. 715).

Voogt et al.'s (2015) investigation, however, acknowledges some of the challenges around the use of the term. Beyond the lack of definitional clarity, they highlight the challenge of being too precise, which would have the effect of restricting it largely to computer science, although a too encompassing-a-definition might render the notion of computational thinking almost meaningless (Voogt et al., 2015). The later observation is significant, because arguably all 'disciplines' have transfer effects. Not too long ago, Australian school education was heralding the transfer effects of learning a language other than English (Fernandez, 2008). Consequently, there is the risk, as Voogt et al. (2015) suggest, that in staking out too broad-a-scope of effect, the specificity of computational thinking as a coding outcome diminishes. Indeed, part of the 'push' for computational thinking in school education is consequently grounded in the claim that this is something that we 'innately' already do (Wing, 2008), which 'may' be correct. But arguably, a key difference between this default computational thinking and that associated with coding seems to be that the latter is mostly actioned through and with the screen, as part of a digital communications network, using (programming) languages which inherently impose limits on how such thinking can take place and be externalized. The computational thinking undertaken by children interfacing with computers is arguably of a different kind, hence possibly inscribed by the kinds of challenges Baudrillard highlights in his discussion of humanity's deepening engagement with (digital) abstraction (Baudrillard, 1993, 1995).

Arguably, it is this aspect of computational thinking (CT)—its unintended consequences for individual and society—that is underexplored in formal literature on coding in schools, beyond what CT comprises in terms of definitions and its relationship to learning outcomes (Gover & Pea, 2013; Shute et al., 2017). The default assumption of much discussion is that because students live in a networked information age, coding and computational thinking are—by default—individual and social goods. Unfortunately, this perspective is seldom elaborated with reference to the political philosophy framing society and the individual's relationship to society for which computational thinking is held as significant. An exception is the work of Bresnihan et al. (2015) that deploys Marxist theory to problematize mandatory coding. Their (sociological) argument is that state and industry support of the coding initiative results from the desire to increase labour supply, thus decreasing coding labour costs.

Bresnihan et al.'s work is significant not because of the veracity of Marxist theory in this application, but because it comprises an instance of critical thinking from the humanities in understanding coding in schools. Though critical thinking is held to be an outcome of computational thinking, as is creative thinking (Shute et al., 2017; Voogt et al., 2015), much research investigating computational thinking for coding in schools is devoid of 'critical thinking', as understood in the humanities more broadly. With this, there is an absence of discussion around the political philosophy framing the society and the individual's relationship to it for which computational thinking is held essential.

A clue as to what kind of society this might be, and what political philosophy frames it, is offered in Knobelsdorf et al. ([2015](#)) discussion. Referencing the relatively early introduction of coding in German schools, they note that the curriculum was labelled with the term *informatische Bildung*. Unpacking the meaning of this term, Knobelsdorf et al. explain that:

> As Meyer [2009] puts it, 'Bildung has no simple, straightforward English equivalent. In Germany, we define a person with Bildung as one with the capacity to participate in his or her society and the willingness to take responsibility for himself/herself and for others. In this view, Bildung is more than education: it is education in a humanistic sense. It aims for self-regulation and includes a moral dimension.' Accordingly, within the German CS education community, the question of how to offer CS education in secondary schools— either as training in ICT skills or aligned with the scientific methods and principles of the academic discipline—has been discussed with regard to the principles of 'Bildung' for more than two decades. From those discussions, the term 'informatische Bildung' (a combination of 'Informatik' and 'Bildung,' roughly general CS education) was coined. The German CS education community argues that general CS education can provide specific knowledge and competencies for students to become conscious and responsible users and makers of information to participate fully in an ever-evolving information society and digital world (2015, p. 7).

Though as we are informed, the notion of *informatische Bildung* 'has no simple, straightforward English equivalent' (p. 7), it resonates strongly with key tenets of liberal-humanism, highlighting the value of coding education to the individual and society. What is more, this relationship is to be understood 'in a humanistic sense' (2015, p. 7), such that *informatische Bildung* can be considered 'akin to American understandings of … liberal arts' (p. 7). References to 'responsibility' and 'self-regulation' strengthen this conclusion, detailing a relationship between coding in schools and a particular political philosophy in ways which much contemporary discussion does not.

While the absence of a more specific articulation of the political philosophy framing the compulsory coding in schools rationale is understandable given that the liberal-humanist underpinnings of modern schooling are perhaps simply a taken-for-granted, arguably, this absence functions to preclude consideration that coding for all might actually produce outcomes inconsistent with key tenets of liberal-humanism. This dilemma is consequently similar to that highlighted in Kissinger's reading of the relationship between AI and the Enlightenment (2018). It is in response to this absence that this discussion of coding in schools aims to move our academic framing beyond a focus on pedagogy, the order of introducing coding skills and technologies and the difficulty of coding. While these remain important, there is arguably a larger framing needing to be invoked to understand the state's rationale for coding in schools, and the popular response to this.

Methodology

The research project from which this paper emerges was concerned to understand the popular response to compulsory coding in schools and how—drawing upon Foucauldian

notions of discourse (Foucault, 1972)—positions within this response construct truths about coding, individuals and society. Based upon the significance of discourse to constructing social truths (Foucault, 1972; Foucault, 1981; Luke, 1997), and the methodological approach of Tamatea in the analysis of populist online discourse (Tamatea, 2005; 2008; 2011), online texts were chosen for a number of reasons as sources of data. Among these is the capacity of the Internet to provide closer proximity to circulation of contemporary discourses. Online discourses can be accessed in (almost) 'real-time', from a larger source of 'participants' (Ungar, 2001). While online 'discourse' communities reveal a shared sense of interest and purpose, the higher degree of (relative) anonymity—a factor currently informing policy discussion around social media regulation (Phillips & Bartlett, 2018)—increases opportunities for personal disclosure, such that online comments about compulsory coding likely reproduces a 'truer' account of feelings (Joinson, 2005). This data sourcing strategy thus follows the recent example of Tamatea and Paramatasari (2018), who collected rich online data to map current approaches to providing coding classes for disadvantaged youth.

Data were generated from a dynamically bounded review of online posts about compulsory coding in education using the Google search engine in December 2017, which allowed the search to respond to emergent trends in results. Terms used for the online search are presented in Table 1.

The search revealed a range of non-formal and grey literature including blogs, news reports, webpages, policy and reports, curriculum documents and associated social media commentary. The search for formal academic literature pertaining to compulsory coding in schools, however, was conducted through an Australian university library using international enterprise data bases.

Critical discourse analysis was used to analyse the online texts, not to verify their truth, but to identify how truths about coding in school are constructed through the process of description, definition and delimitation (Sherridan, 1980). As Altheide (2003) (drawing upon Schwalbe) highlights, discourses are more than just talk and writing. Through description, definition and delimitation, discourses impose rules about what can be said and written and who has the authority to say and write things accepted as truth, which regulate and maintain social action and boundaries.

The initial criteria for analysis of the online texts were simply the text's position in relation to the question: Did the text support or oppose coding in schools? Supporting texts were classified as belonging to the 'Yes case', while opposing texts were associated with a 'No case'. Approximately 135 textual artefacts were initially identified, representative of the Yes case. Forty No case artefacts were identified. Working with these two discursive orientations, a table was constructed into which arguments deployed by each case were inserted. These arguments were, with few exceptions, mostly common to

Table 1 Initial Google search terms

Search terms
Computational thinking
Criticism/critique/problems of computational thinking
Coding movement
Compulsory/mandatory coding in schools
Coding literacy

both cases. They were subsequently coded into a range of equally common themes, as shown in Table 2.

While critical discourse analysis (CDA) is not a monolithic project, multiple concepts and tools can be mobilised under the banner of CDA; it is important to appreciate that discourse comprises more than simply words, topics or themes. Rather, as noted, discourses comprise socially shared patterns of language (as texts). They are neither random nor neutral but deployed strategically in the social contest of truth construction, linked to material outcomes (Foucault, 1981). A macro-level analysis explored how the arguments for and against compulsory coding are both produced by and seek to reproduce versions of the arguably dominant political-philosophy framing education circulating within the broader socio-political context in which the debates occurs. In this, a key question was: Why out of all the things that might have been possible to say about compulsory coding in schools have only certain things been said?

Finally, the author is an education sociologist and developer working with C#, JavaScript and the NET environment more broadly. I declare this because in this author's experience, computer science and sociology can make 'strange' partners. Though both deal in abstraction, one is more qualitative and hence seen as 'subjective' while the other is more quantitative and seen to be more 'objective'. Because of this, these disciplines can uphold antagonistic claims to truth. However, lest this author be viewed as favouring one side or another, or failing to understand one or the other, it is

Table 2 Discursive themes

Shared themes	Yes case-only themes	No case-only themes
• The purpose of learning and education	Biological/brain	Esoteric/little use
• Moral concerns	Fear	How I really learned programming
• Overstating the easiness of learning to program	Fix own things	Not everyone can code
• Personal/individual benefits	Urgency	Screen time
• Economic benefits		
• Ubiquity of computers and coding		
• Ability to fix one's computer-related problems		
• Thinking skills and creativity		
• Computational participation		
• Career benefits		
• Crowded curriculum		
• Historical parallels		
• Student ages		
• Spill-over/transference		
• Technologies/languages		
• Other school subjects		
• Social justice		
• The times		
• Technology creators		
• Literacy		
• Engaging artificial intelligence/ computers/robots		

perhaps best to declare one's position and acknowledge that in some instances, objectivity emerges from the recognition of our own subjectivity (Schatzki, 1995).

Social theory

Analysis and discussion below is framed by social theory pertaining to liberal-humanism and Baudrillard's argument around intensifying abstraction (Baudrillard, 1993; Baudrillard, 1995). While these framings are unpacked in detail later, it is valuable here to briefly introduce their 'fundamentals'.

Liberal-humanism

Liberal-humanism is grounded in liberalism and humanism, the former historically preceding the latter. Humanism comprises a conceptualization wherein humanity occupies the centre of its own existence. This once 'radical' framing holds that humans are capable of rationally knowing their world, and its workings, and that such can be explained without referencing deities or non-physical properties (Agar, 2012). Humanism has thus been grounded in a knowledge of humanity and the material world realised in the Arts and Humanities, and the Sciences (Good, 2001). The 'modern' West generally attributes its understanding of humanism to the Renaissance, its associated literary tradition and the Enlightenment (Good, 2001; Kircher, 2015; Yousefi, Yousefy, & Keshtiaray, 2015).

Whereas humanism might be considered a metaphysical view of humankind, liberalism, arguably emerging from humanism, comprises a political philosophy (Gahringer, 1995). Liberalism is concerned with humanity's existential status, particularly the individual, and how socio-economic factors are arranged to uphold individual and social liberty. Liberalism is thus concerned with the distribution of goods and rights (Riberio, 2014). Although there are streams of thought and practice within liberalism, a common 'thread' comprises a commitment to individual liberty and freedom, the equality of all, and (broadly) process (es) leading to their achievement. Liberalism holds that individuals have an inherent right to pursue the 'good life', a consequence of reviewing a range of conceptualizations of such. The decision to pursue one life over others is an exercise in individual liberty and freedom, the outcome of rationality and reason grounded in deliberation upon the 'facts'. Knowledge (as opposed to ignorance) is central to this (Singh, 1998; Titlestadt, 2010). The individual subject, truth and factual knowledge are thus important to liberal-humanism and the Enlightenment more broadly.

A key difference within liberalism concerns the extent of (state) regulation to secure liberty and freedom. Those holding to positive liberty accept that while individuals have an inherent capacity to be rational, this might not be exercised if one is not equipped with the knowledge to foster its emergence. Hence, regulation is required to prevent individuals infringing the freedom of others and society more broadly, as much as for their own interests. By contrast, negative liberty holds that the individual must be free to establish *themselves* what counts as valuable knowledge and that the state has the right to determine what this should be (Guilherme, 2016; Singh, 1998; West, 1965). Making coding compulsory in schools represents an instance of positive liberty.

Baudrillard and abstraction

> It looks so ridiculous and impossible, Musk told reporters in a press conference after the launch. You can tell it's real because it looks so fake, honestly (Geggel, 2018).

Elon Musk, described as inspiring a new generation of engineers and space explorers (Keenihan, 2017; Rogers, 2018), made this statement after launching his red Tesler Roadster into space (Matousek, 2018). It references imagery of the car subsequently streamed to Earth. Predictably, online discussion asserted it was fake. YouTube videos claimed the roadster was never launched; it was simply digital trickery. The dilemma with these claims, however, is not that we can never really know if the car is in space, because most cannot access the referent. Neither is it because the YouTube like other social media platforms is used to spread 'fake news', challenging the veracity of the Enlightenment notion of the free market of fact-based ideas (Schroeder, 2018). Rather, it is because with powerful digital imaging tools grounded in abstraction, it may well be—just a simulation (Leary, 2017; Vincent, 2017b).

This dilemma signifies our times, characterised by Baudrillard as those of the third phase of the image (Baudrillard, 1995). For Baudrillard, arrival at this point, wherein interaction with the world, self, others and 'society' is increasingly mediated by a digital abstraction layer, has proceed through three orders, each grounded in a relationship between humanity, the representational image and reality (Baudrillard, 1995). The significance of this, to the analysis of the data below pertaining to the Yes and No cases, is not only the resonance between popular argument and Baudrillard's stages of the image, but also these orders provide a thought framework for reflection upon the implications for liberal-humanism arising from this resonance. However, with the first order grounded in dissimulation (Baudrillard, 1995, pp. 2–4), it is, then, the second and third that are of more relevance here. In the second order of simulacra, a shift takes place, from dissimulation to simulation, the move to simulate or 'feign(s) to have what one doesn't have' (p. 2). In the second order, roughly from the Renaissance to mid-twentieth century, industrial technologies impact image production, including (analogue) photography leading to mass production and re-production of copies. Still, a difference remains between the real and copy. An analogue photo cannot be mistaken for the real.

By contrast, the third order is 'founded on information, the model, the cybernetic game – total operationally, hyper-reality, aim of total control' (Baudrillard, 1995, p. 79). This is our time, one in which the analysis of the data suggests notions of the rational liberal individual are grounded in the capacity to think in the abstract language of the computing machine and be 'enmeshed' in its broader network for collaborative social participation. In this order, we increasingly inhabit digital spaces that are not real (in an analogue sense) and yet not fake. They do not necessarily aim to copy the real, as much as provide some other kind of (virtual) space within which to interact. In this order, the digital object may have 'no relation to any reality whatsoever' (Baudrillard, 1995, p. 4), being pure abstraction and simulation. And when reabsorbed back into the construction of the real, it creates a hyper-reality (p. 8) wherein reality engagement is increasingly via digital models against which the analogue reality (including people) is evaluated. Simulations reference simulations, while the corpse of the real lies 'rotting' in the corner (p. 1). With the intensification of abstraction, the territory no longer precedes the

map, nor does it survive it—'welcome to the desert of the real'—as Morpheus (drawing on Baudrillard, 1995) explains to Neo in the Matrix (Warner Brothers, 1999). Today, Baudrillard asserts abstraction is increasingly the *only* means through which we access the real, the outcome of a logic which perhaps only ends through implosion or 'the destruction of meaning and the reality-effect' (Robinson, 2012). Simply put, humankind increasingly prefers the map (abstraction) to the territory (which is the real). Coding in schools seems an exercise in how to best engage the map.

Results

Data analysis identifies three dominant discourses in the popular response to coding for all: the digital ubiquity, the disadvantage and the habits of mind discourses. But before data informing these discourses are identified, it is worthwhile noting the relative 'strength' of the discursive themes in terms of the number of individual expressions or arguments within broad groupings deployed by each case. In terms of themes shared by both cases, among the strongest are the economic, personal benefits, thinking skills/ creativity, technologies/languages, social justice, literacy and engaging AI/computers/ robots. Within these, the Yes case dominates in terms of the number of arguments articulated in all except for the social justice and AI/computers/robots themes. In terms of themes deployed by the Yes case alone, a number of these were relatively weak in terms of the number of individual arguments deployed. By contrast, several of the arguments within the No case alone themes were relatively strong. These included overstating the easiness of learning programming, and coding in schools being too esoteric and of little use. However, other than these perhaps 'outlier' themes, the majority of the arguments in the popular response are contained within broadly shared themes, though argued from oppositional positions. This coalescence and coherence of argument around strong themes informs the dominance of the discourses identified in this debate.

Digital ubiquity discourse

Both sides in the debate deploy a digital ubiquity discourse. The Yes case maintains the digital impacts all aspects of our lives (Ruseva & Rissola, 2016), and that computers are pervasive (Kafai & Burke, 2014, p. iv), while the No case acknowledges this is 'an increasingly tech-driven world' (Editorial Board, 2017, para 1). Argument diverges, however, in relation to providing evidence supporting the representation of digital ubiquity and in constructing what digital ubiquity implies for education.

Regarding evidence, the Yes case (like the No case) is grounded in sources of textual authority, citing industry reports, government data and authoritative voices. We are told that 'six of the world's ten most valuable companies are technology companies, with Apple leading the pack' (Broadband Solutions, 2016, para 1). We are also told that according to the Australian Workforce Productivity Agency, 'in 2025 there could be an undersupply of qualifications for key ICT occupations, with employment projected to grow between 64 and 72 per cent faster than overall employment growth' (Carroll, 2016, para 3). This argument references Australian Chief Scientist Ian Chubb, stating that 'if the digital economy is the arena, then the skills you need to play include computer programming and coding' (Carroll, 2016, para 5).

Consequently, the Yes case constructs the economic value of learning to code in terms of the digital economy and its skills requirements. Cohen, for example, reproduces statistics from Gartner showing that globally in 2015, IT spending would not only reach $3.5 million, but use of digital technologies would 'grow the size of the global GDP by $1.36 trillion in the next 5 years – all of those dollars equal opportunity for the people who are fluent in tech' (Cohen, 2015, para 2).

But as the reference to Gartner suggests, the Yes case represents the economic need and employment opportunities as not only of national significance, but also international. Puiu notes that 'according to the European Commission, there will be a shortage of over one million programmers in Europe by 2020' (European Commission, 2016, para 1), while according to Mayo, 'in the US alone there'll be a million more computing jobs than computing science graduates by 2020' (Mayo, 2013, para 1).

For some in the No case, the economics argument is not so much an opportunity for workers as for 'capital', as argued in the literature by Bresnihan et al. (2015). Silicon Valley, it is claimed, 'has been unusually successful in persuading our political class and much of the general public that its interests coincide with the interests of humanity as a whole' (Tarnoff, 2017, para 12). Tarnoff adds, 'it will proletarianize the profession … flooding the market and forcing wages down – and that's precisely the point' (para 4). The relevance of this argument is somewhat validated by Yes case discourse, not in the sense of a capitalist conspiracy, but in recognition that an increased supply of developers will lead to lower wages. Yes-case proponent, McAllister (2008), suggests that 'computer literacy should be seen as a baseline skill for the U.S. workforce, not a differentiator' (para 20) leading to a 'ticket to a golden future' (para 20).

Referencing economics, the Yes case also highlights ethno-national divisions as justification for compulsory coding. This boundary constructing position references fear and danger, expressed through the notion of crisis and risk (Bagshaw, 2015; Mayo, 2013). In the Australian context, Carroll (2016) highlights the 'global nature and ferocity of the competition' in software development (para 5). Others maintain that without 'compulsory' coding, there is a risk that 'Australia [will be] left behind in the digital age' (Calixto, 2015, para 1). Regarding the USA, Bajarin (2014) notes that it is 'far behind in having a robust technical workforce created within our own boarders' (para 13). Reference to borders and 'us' and 'them' is significant in the Yes case. Regarding innovation, we are told that coding in schools will ensure that the Microsoft and Google continue to be grown 'at home', in the USA (Cellan-Jones, 2014). Elsewhere, 'we' are warned of reliance on 'foreign' workers, including those from 'a variety of Asian companies' (Debate.org, 2013, para 3), and specifically 'coders in China, India and other parts of the world' (Bajarin, 2014, para 12).

The Yes case's deployment of the ubiquity discourse further represents compulsory coding as an individual and social good. Lexical items such as 'empower' (European Commission, 2016, para 4) and 'enhance' (Lucisano, 2017, para 2), construct the value coding skills add to the individual. Compulsory coding will enable individuals to understand 'how their devices work' (Bajarin, 2014, para 9), allowing them to fix them (Hinsliff, 2015). It will allow individuals to know 'how [digital technologies] operate and how to customize them for better functionality' (Lynch, 2017, para 3). Coding, Enobrev argues, will help students gain a 'better understanding of the world and hopefully better interact within (sic)' (Atwood, 2012, para 3). Agon likewise associates coding with how to better 'live in this new economy and society' (Rawlins, 2015, para 5).

By contrast, the No case asserts, 'no matter how pervasive a technology is, we don't need to understand how it works' (Felker, 2013, para 9). Constructing less relevance between digital ubiquity and education, the No case highlights the division of labour and employment specializations. Argument maintains that as the employment sector is characterized by different people doing different jobs, the need to understand how code works in order to fix something is mitigated by the existence of those whose job it is to fix such code. As Felker notes, 'society divides it labour so that everyone can use things without going to the trouble of making them' (para 9). A further argument advises the reader not to overlook that society still requires a broad spectrum of employment fields.

> We're always going to need doctors and nurses, teachers and chefs, all sorts of jobs that don't need to write computer code at all. To assume that everyone should be required to learn it is to be caught in the bubble of your own profession (Hall, 2017, para 8).

The 'ubiquitous though less relevant' argument proceeds by highlighting the extent to which ubiquity in other fields does not mandate subject specialization. Parallels are drawn with music (Rawlins, 2015), motor vehicles (Atwood, 2015; Gray, 2014), plumbing (Atwood, 2012), carpentry (Bethune, 2016), aircraft (Gray, 2014) and electrical work (Cohen, 2015). Stucky (2015) notes that he does not use any of his coding knowledge to shop or bank online, suggesting that:For the majority it's probably more like understanding a car engine. You don't need to be able to strip an engine and rebuild it to drive a car, but if you basically understand how a car works you can drive and maintain it efficiently and effectively (para 9).

Disadvantage discourse

Data show that argument in disadvantage discourse is more explicitly grounded in liberal notions around resource distribution (Riberio, 2014). The Yes case is grounded in a Rawlsian liberal-distributive framework, the No case in a more libertarian position. Both, however, assert disadvantage. The Yes case maintains those without coding skills are, and will remain disadvantaged, without state intervention. The No case maintains that state intervention disadvantages those already equipped with skills. Both argue from a position grounded in pursuit of liberty.

At its broadest level, the Yes case challenges a (claimed) elitist image of software development, on grounds that it is uninviting to the underrepresented masses (Raja, 2014). Compulsory coding, it is argued, will demystify programming (Rawlins, 2015). A challenge to 'elitism' is also expressed through re-asserting the value of amateur coding as opposed to professional. Neil argues that 'it skirts close to snobbery to discourage somebody merely on the grounds that it isn't their day job' (Atwood, 2012, para 73).

Specifically, the disadvantage discourse foregrounds equalizing access as the principal justification for compulsion. Compulsion, it is argued, will prevent disadvantage arising from some schools implementing coding and others not, and boys choosing coding and not girls. Compulsion will equalize access to a field often unavailable to or not accessed by underrepresented students (Della Cava, 2015; Gilbert, 2017), exposing their talent

(Guzdial, 2014). As AnonymousCoward, 2012, para 53) notes, 'it is a meritocracy, and it should be - but if you don't look for talent, you won't find it all'. In sum, as argued by Lyonnais, compulsory coding might 'level the playing field' (Lyonnais, 2015, para 8), particularly in terms of gender (Gilbert, 2017). Compulsion will allow the underrepresented to exercise choice about the relevance of coding to later school and post-school life (ECDL, 2015).

Like the Yes case, the No case deploys a disadvantage discourse across several fronts. One concerns existing inequalities, another concerns likely inequalities, and yet another responds to (in) equality arising from state intervention in the coding 'market'. With regard to the former, Keneally (2015) asserts there are unresolved matters in school education more significant to improving student outcomes. Citing Krugman's commentary, Keneally reaffirms that while 'talking breathlessly about how technology changes everything might seem harmless … [its] a distraction from more mundane issues' (para 11), adding:

> that includes issues such as failing schools, poor teaching quality, lack of specialist support for students with a disability, the increasing shift away from comprehensive publicly funded schools to private education, poor international rankings in literacy and numeracy, and countless other real, and very difficult issues, that confront Australia's education system (2015, para 12).

Keneally concludes that 'sorting out those problems will take much more than a few lines of code' (2015, para 13). Here, compulsory coding perpetuates existing disadvantage. Elsewhere teaching quality is highlighted. The demand for skilled coding teachers, which may not be met, will result in unqualified teachers delivering the curriculum, resulting in a watered-down curriculum (Guzdial, 2014, para 19). Guzdial (2014) adds, regarding inequality in terms of access, 'let's work first at making it accessible, before we try to require it' (para 19).

The No case references the 'crowded curriculum', constructing a concern for teacher welfare (Debate.org, 2013; Dodge, 2017) and asserting the greater value of other subjects to the liberal-humanist project of dispelling 'ignorance'. While this argument does not contest the role of the state in providing children with access to knowledge supporting liberty, it represents coding as deleterious to the capacity of 'more valued' subjects to achieve this goal. At Debate.org, a post in the 'con' argument asks: 'what subject will you be robbing these children of, Science? Math? History, or the already underfunded art and music departments?' (2013, para 6). Aside from representing compulsory coding as robbery, argument highlights the desire of some in the Yes case to have humanities subjects make way for coding, such as Geography (Lui, 2015), Religion (Debating Europe, 2015), or History (Lui, 2015). In Lui's (2015) article, Dunford, for example, asserts, 'give the boot to art/woodwork/cooking and lessen the time for sport' (para 1), though he supports retaining Geography and History. Elsewhere, languages would be removed (Hait, 2017).

The No case equally acknowledges the significance of choice; however, its reference is elaborated in terms of the child's right to make their own choice about subjects beyond the 'basics', in which coding is to be included. Capone, for example, asserts regarding mandatory coding, 'no, this isn't the right way' (Debating Europe, 2015, para 1),

arguing that elementary schools should provide the basics, and leave more specific areas to specific choices later on (para 1). This argument, highlighting student age, holds that younger students are disadvantaged through deprivation of the option to choose coding later in life, when more capable of making informed choices. Others, however, assert the right of parents to make choices about their child's school subjects (Debate.org, 2013).

Argument deploying references to 'choice' also highlights likely outcome of compulsion—coding enjoyed by few and hated by many (Guzdial, 2014). Mainstreaming coding to generate equality may result in unequal distribution of the enjoyment of coding. Dkramer3 adds, 'if you want kids to hate programming … make it mandatory' (Haaramo, 2015, para 19). This argument represents not only the remainder of the 'mainstream' as potentially disenfranchised, but also students of lower academic ability. This too is a concern with disadvantage. Dillon asks: 'do we know how to teach CS to students with learning or development disabilities? Can we confidently state that, without CS, those students should not earn a high school degree?' (Guzdial, 2014, para 7). Consequently, the No case represents compulsory coding as yet another opportunity for students to fail, creating a two-tier subject performance cohort (Computing Education Research Blog, 2014). What is more, the No case argues that the level playing field will be distorted by the power of economically advantaged families to purchase support resources beyond those provided by the state (Editorial Board, 2017).

The No case maintains that a commitment to coding should arise naturally through individual passion (Chris-Granger.com, 2015) and intelligence (The Register 2012). While acknowledging one is not a Luddite if one cannot code (Chris-Granger.com), many in the No case assert that coding is neither easy nor quickly learned, such that to assert otherwise devalues the profession and intelligence of those who can code (Byrne, 2013). Hence, the No case holds that anti-elitism disadvantages existing coders. As Hartnell explains:

> I hate the current railing against 'elitism' from the know-nots. They would be the first to complain if their football team didn't field elite players, or their medical treatments weren't by highly-skilled professionals, but they also feel threatened by the intelligentsia, so try to drag everyone down to their level (Young, 2012, para 100).

Habits of mind discourse

Whereas data show a ubiquity discourse constructs compulsory coding as enabling individuals to engage digital technologies at a generic level, the 'habits of mind' discourse deployed by the Yes case constructs engagement as both cognitive and cerebral. By contrast, the No case reasserts a need for engagement with the analogue. Indicative of the Yes case, Vikberg argues that programming needs to be integrated across the curriculum 'to drive critical and creative thinking as computer programmes have an increasingly significant part to play in the society today' (Haaramo, 2015, para 3). Others assert, 'we need to remember that computer science is a creative field in which students are actively creating something' (Team ISTE, 2015, para 9). Hai Hong from Google's K[-12] outreach programs maintains that 'it offers problem-solving skills and promotes creativity' (Della Cava, 2015, para 7). Elsewhere, a School Education Gateway news report informs that 'learning to code also develops skills such as

problem solving, [and] logical reasoning and creativity' (2015, para 3). And beyond creativity (ECDL, 2015), a post at Familytech (2017) highlights coding's capacity to improve 'essential life skills like critical thinking, problem-solving, and creativity; which can lead to success in other areas of work'. Creativity, critical thinking and reasoning figure strongly in this (Yes case) discourse.

This representation constructs coding's 'spill-over effect'. Coding is good not only because it helps create programs, but also because its skillset can be used elsewhere. Though precise details of where else are often absent, the comment of Garun, reproduced by others such as Team ISTE (2015) and debate.org (2013), represents this argument.

> Learning to code contains the same logic skills you apply in daily life: What is the problem? How can I solve the problem as efficiently as possible? Can my solution be helpful to others who are experiencing similar issues? If you can figure out the same steps from a programming perspective, it can help develop your logic and decision-making skills to streamline the best solution to your problems (2013, para 14).

Like other discourses, this too appeals to authority, in this instance, to well-known software developers, 'research' and education policy. Concerning the former, Bill Gates (Microsoft) and Mark Zukerberg (Facebook) are cited in terms of 'code' and creativity (Burks, 2017, para 5). Steve Job's comments about programming and thinking are also reproduced by the Yes case (Puiu, 2017). Appealing to the 'authority' of research, we are told that 'numerous formal studies have confirmed coding has measurable beneficial effects on cognitive abilities' (Rawlins, 2015, para 18). Not only are state curriculum and policy documents referenced, so too are texts detailing educational principles and philosophy (Computing Education Research Blog, 2014).

The 'habits of mind' discourse is, however, also strongly grounded by the Yes case in the notion of computational thinking. As the president of the Australian Computer Society (ACS), Ansley argued that computational thinking should be taken 'more seriously' in schools, maintaining 'it's the fourth 'r' … three 'r's plus coding, or computational thinking' (Calixto, 2015, para 19). Elsewhere, a Queensland Government (2015) paper identified coding as a twenty-first century literacy (p. 5).

This computational thinking framework is extended by the Yes case to include computational participation, emphasising networks, collaboration and the social (Kafai & Burke, 2014; Queensland Government, 2015). An EU text, for example, likewise asserts that, 'each and every interaction between humans and computers is governed by code', such that in our 'hyper-connected world … coding is the literacy of today' (European Commission, 2016, para 2). Others, however, reference the human brain, with Lopes arguing that, 'it's a matter of development of the human brain, to increase logical skills and prepare for the future of cyberworld' (Sololearn, 2017), a (biological) framing reproduced in discussion at Broadband Solutions.

> Like learning a language early in life, learning and practicing this type of [programming] thinking early in development actually influences a child's brain as it is still developing (Broadband Solutions (2016), para 7).

By contrast, though deployment of the habits of mind discourse by the No case is not so focused on the cerebral, it engages similar discursive tactics. It appeals to authority, speaks (back) to the 'spill-over' effects and engages the impact of computational thinking in terms of creativity and critical thinking.

Appealing to the authority, the No case references Jeff Atwood, co-founder of *Stack Overflow* and *Stack Exchange*. Saines (2014) explains, for example, that 'those defending the need to teach young children to program don't have a solid counter-argument when luminaries like Jeff Atwood say that not everyone should learn to program' (para 4). Indeed, Atwood (2012) himself maintains that 'the whole everyone should learn programming meme has gotten so out of control that the mayor of New York City actually vowed to learn to code in 2012' (para 1). Other references delegitimise 'authority', through defining the authority to speak about education. Commenting on the views of the Apple CEO, Pewen and Shotton represent Tim Cook (and Bill Gates) as unqualified to speak about education. Pewen asserts that beyond Cook 'making [ing] the most idiotic statements on a regular enough basis', he is gravelled by Cook and Gates 'shooting off about education it's not your field' (Hall, 2017, para 16).

In response to claimed spill-over effects, Sandy replies, 'try teaching a few dozen computer science classes and see if you still think CS is better prep for life skills than *any* (sic) other subject. It's not' (Rawlins, 2015, para 21). Sandy challenges the veracity of the 'research' cited as evidence in terms of creative thinking, asserting, 'it is a myth. You can say the same for creative writing, but neither should [it] be compulsory' (2015, para 19). This is an interesting response, not because all subjects might claim spill-over effects, but because it interrogates the positive link between computational thinking, creativity and critical thinking constructed by the Yes case.

By contrast, the No case foregrounds a more humanities-based understanding of creativity and critical thinking. At one level, this emerges in referencing the significance of humanities-based knowledge to the digital. Software designer, WiscoNative, explains that his 'useless' sociology degree taught him much about social interaction, meaning that he could 'design better software that users can find more intuitive' (Hall, 2017, para 16).This comment contests the primacy of quantitative thinking to creativity by constructing a link with humanities in this domain. Indeed, in claiming we have gone too far with the STEM agenda, Dumdum argues, 'let's not forget that many of our values and cultural developments, from democracy to the Enlightenment, actually came from times and places where the Liberal Arts ruled' (Hall, 2017, para 14). In contrast to the networked minds of the Yes case, the No case maintains that children should first be taught to play outside and read books (Debating Europe, 2015). What is more, the No case argues that children should be 'developing human skills … running around, throwing a ball, scratching out drawings, learning fine motor skills and developing normal interpersonal skills' (Dvorak, 2014, para 2).

Discussion

Data analysis above highlights three dominant discourses deployed in debate around compulsory coding in schools, including the *digital ubiquity*, the *disadvantage*, and the *habits of mind* discourses. Despite deployment to achieve different outcomes, there is a (qualified) degree of resonance between protagonists in this debate. At one level, this emerges from the use of common discursive themes (see Table 2). At another level,

however, it emerges from both cases validating broad principles of liberal-humanist philosophy, which variously underwrite modern Western education (Crittenden, 2006). Thus framed, these cases either deploy a negative conceptualization of liberty (the No case), or a positive conceptualization (the Yes case). With the exception of elements of libertarianism in the No case, both remain comfortably liberal. Notwithstanding this, there are aspects of the habits of mind discourse in particular that are potentially destructive of liberal-humanism, the individual subject, liberty and freedom as traditionally constructed. Below, discussion critically reviews these discourses, identifying the resonance with liberal-humanism and the broad orientations within this, namely positive and negative liberty.

Liberal-humanism and digital ubiquity

Deployment of the digital ubiquity discourse by the Yes case seems convincing. Facts, figures and authorities confirm reliance upon the digital. Coding is as essential for self-efficacy and social participation, an individual and social good. With this, the Yes case reproduces long-established views of the relationship between education, the individual, and society held by positive liberalism (Crittenden, 2006; Guilherme, 2016), many traceable to debate around the 1831 Reform Act in the UK, focused upon compulsory schooling. In this, Roebuck argued that compulsory education would protect the well-being of individual and society. A compulsory state education 'would teach people how to be happy and therefore would reduce violence, mischief and political unrest' (West, 1965, p. 134) and enable people to understand their circumstances and improve them (p. 134), reducing the number of the 'stack-burning peasantry' (p. 134). Moreover, compulsory education would enable England to remain internationally competitive (p. 134), rationales similar to those articulated above by the Yes case and for introducing coding in the German context noted above [van Weert 1984] (Knobelsdorf et al., 2015).

   In this historic debate, the opposition raised concerns resonant with the No case's response, including questioning which knowledge and whose knowledge informs the curriculum and its relevance to those compelled to learn it. Opposing compulsory education, Godwin rejected state power to decide what people should learn. Compulsory education, he maintained, was 'only too easy a channel for thinkers who were arrogant enough to believe that they had the monopoly on the truth and that their doctrines alone were worthy of forced consumption through the agency of the state' (West, 1965, p. 31). Opposition to UK compulsory education highlighted, like the No case, the possibility that the state might not always act in the people's interest (West, 1965). Suspicious of state intentions, Godwin saw a potential for despotism and corruption (West, 1965, p. 131, p. 133), a suspicion reproduced by the No case regarding developer wages. Lest it be thought that laisse-faire liberal John Stuart Mill would have opposed UK compulsory schooling, he actually revised his negative stance on liberty in this matter (p. 137).

Liberal-humanism and disadvantage

Although those like Godwin criticised positive liberals for their elitist imposition of values (West, 1965, p. 131), in the compulsory coding debate, it is the Yes case (positive liberty) that, to contrary, seeks to mandate coding to challenge elitism and

disadvantage. Extending coding skills to all, the Yes case is seemingly grounded in Rawlsian (liberal) distributive social justice, wherein disadvantage is reduced through state resource re-distribution (Rawls, 1999). Although the No case is also somewhat inscribed by this view, it offers two counterarguments. The first replays the redistributive logic but argues existing resources are insufficiently redistributed, a call for positive liberty. The second is more clearly articulated from a position of negative liberty, highlighting the futility of state resource re-distribution. This argument asserts the power of the market to negate state intervention. It maintains that students' differential access to other resources will result in perpetuation of the gap between the already well-resourced in terms of coding skills and those not.

The No case is also framed by elements of libertarianism, arguing the removal of the state altogether from resource redistribution to redress disadvantage. The sub-text being a disadvantage is often the result of individual action and not that of either others or the state (Coleman, 1976), such that redistribution interferes with the economic market, and the market in 'natural' (coding) talent and motivation (Coleman, 1976, p. 128); argument resonant with Nozick's Libertarian rights-based 'inequality' position (Coleman, 1976, pp. 121–122). Beyond this, the No case's assertion that compulsory coding will either produce a 'class' of students who can code, as opposed to those who cannot, who will subsequently 'hate' coding, further resonates with the experience of compulsory coding in earlier implementations (Knobelsdorf et al., 2015).

Habits of mind, abstraction and a loss of self, liberty and the social

The Yes case seeks a level playing field providing students with choice to take up coding as a career—or not (Rawlins, 2015). The No case maintains, however, that children are too young to engage coding skills, such that coding should be offered later in scholastic life as an elective (Debating Europe, 2015). It further asserts the right of parents to make this choice (debate.org, 2013). Key issues here concern age and parental rights, both of which not only figure in historic education debate (West, 1965), but continue to exercise liberal thinking, particularly in liberal democratic state institutions challenged by expressions of multiculturalism and religious freedom (Deagon, 2018). Often (though not always), courts uphold the rights of the state over parents from religious groups and intentional communities in curriculum opt-out cases (Good, 2001; Mouritsen & Olsen, 2013; Singh, 1998), on grounds that parents may not be in the best position to foster the child's capacity to choose their version of the preferred life from a *full* range of the alternatives.

While these rulings and subsequent academic commentary (Good, 2001, Mouritsen & Olsen, 2013, Singh, 1998) highlight the significance liberal-humanism accords habits of mind (reason and rationality as pre-conditions of choice), the Yes case often restrict alignment of these with a quantitative *weltanschauung*, realizable through computational thinking. By contrast, the No case deploys a more 'human-centred' framing, referencing Liberal Arts, Enlightenment values and the Humanities. At one level, this difference is not overly significant. Liberal-humanism (and liberal education) has long been grounded in the Arts *and* Sciences, qualitative and quantitative, and the study of 'humanity' and 'nature' (Crittenden, 2006; Good, 2001; Repp, 2000; Rwodzi, 2014). Yet at another, if Baudrillard's argument around the consequences of abstraction holds

true, the quantitative reading of humanity associated with the Yes case may be corrosive of key elements of the liberal-humanist project including the individual subject, liberty, the social, and dispelling ignorance as a precondition for the realization of these tenets.

Both computational thinking and participation emphasize a relationship between coding and the social. The former's goal is to produce 'creative' solutions to social (among other) problems through coding; the latter's is using coding to access the social and partake in the sociality of coding (Kotsopoulos et al., 2017; Fields, Kafai, & Giang, 2014). As Kafai asserts, 'computational thinking and programming are social, creative practices' (2016, p. 27). For Baudrillard, however, while the social has been declining since separation from the political, its demise accelerated with the onset ICTs, with negative consequences for individual subjectivity, liberty and the social (Baudrillard, 1983; Baudrillard, 1993; Baudrillard, 1995).

For Baudrillard, communication, as opposed to symbolic exchange and the speech act, is destructive of subjectivity and the social to the extent it is operational (staged). Communication is manufactured through imposing network architectures and the quantitative logic of performativity, or 'the compulsion to operationalism' (Baudrillard, 1993, p. 49). Communication is 'a matter not of speaking, but of making people speak' (p. 46). It 'involves not knowledge, but making people know' (p. 46). Thus, participation (the goal of the computational participation) is 'not an active or spontaneous social form' (p. 46). Rather, it is something 'always induced by some sort of machinery or machination' (p. 46). In this, operations as opposed to actions are regulated such that communication is 'operational or it is nothing' (p. 46).

At this point in history, humanity is increasingly tied to this machinery of communication, which both cases acknowledge. While this occurs physically in relation to the screen, the connection is also cerebral, which the Yes case maps. This may have implications for thinking, beyond the familiar narratives around too much screen time. Indeed, for Baudrillard, there is no better model of the way in which the computer screen and the mental screen of our own brain are enmeshed than in the mobius topology (Baudrillard, 1993, p. 56). Though he speaks to the spectacle of thinking in the context of AI, he is equally critical of the impact of digital technologies in the form of computers and programs more broadly (Baudrillard, 1993, p. 53), in relation to which 'telecomputer man offers the spectacle of his own brain' (p. 54), while the analogue dimension of our humanity and subjectivity recede in significance. The Yes case, it will be recalled, deployed a biological argument referencing the brain, while the No case asserted the primacy of analogue experiences. Baudrillard asserts that 'where human relations become mediatized and computerized, as occurs in computational thinking and participation, we interact without touching each other' (Baudrillard, 2009, p. 17),'interlocute without speaking to each other' (p. 17) and we 'interface without [really] seeing each other' (p. 17). In other words, we interact through digital layers (of ones and zeros), where the analogue interlocutor is digitized. Thus, humans become permeable to all images and,

> what gets lost in this new ritual of [information] transparency and interaction is both the singularity of the self and the singularity of the other. … the irreducibility of the subject and the irreducibility of the object (p. 18).

For Baudrillard, communication via the screen emerges from a different paradigm, wherein 'the other, the interlocutor, is never really involved', thus erasing the capacity for intimacy (Baudrillard, 1993, p. 54).

What is more, the screen allows for the disappearance of the other as our gaze upon their digital re-presentation permits reading them on our terms alone. Baudrillard likens this to a digital Plato's cave wherein the other's reality can only be discerned through their digital shadow. 'Why speak when we can communicate?' (Baudrillard, 1993, p. 54). In the screen, we consume as opposed to decipher 'in any depth … information intended to be explored instantaneously' (p. 54), which comprises an 'abreaction immediate to meaning, a short-circuiting of the poles of representation (p. 54). In this model of "interfaces and duplication, of contiguity and networks' (p. 54), two become one (a copy) in contrast to the exercise of the imaginary in the pre-ICT world, where the one was divided into two, as 'governed by the mirror … by otherness and alienation' (p. 54).

Tactility, an essential element of the analogue experience, as argued by the No case, becomes little other than 'epidermal contiguity of eye and image' (p. 55). Analogue aesthetics of distance and time collapsed as a function of simply looking (p. 54). Whereas once the physicality of distance informed production of meaning, with the screen this is gone. In its place a 'perpetual void that we are invited to fill' (p. 55)—although it cannot be filled. What is more, this screen image is equally distant such that it can never be reached through the body (p. 55). Unlike the mirror in which the image always seems more distant, the screen will never allow one to get to the other side. To the extent that the body (as opposed to a disembodied brain in a vat) is a presumably *also* a locus of our humanity and fundamental to authentic exchange, our relationship with the screen places humanity in a space 'that is no longer quite human' (p. 55). Thus, Baudrillard asks: 'am I man or machine'? (Baudrillard, 1993, p. 58). As opposed to the industrial machines of the second phase of the image, in relation to which the individual could 'at least reattain the precious status of alienated man' (1993, p. 58), this is no longer possible. Increasingly, the subject is enmeshed with (computing) machines, as we offload cognitive capacity to 'superior' digital entities, and through computational thinking, our categories of thought become those of the abstraction-based coding language we work with. Wing asserts, for example, that:

> Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines (Wing, 2010 p. 1).

For Baudrillard the question of liberty no longer makes sense in this mediated context (Baudrillard, 2009, p. 19). 'Communication man is assigned to the network in the same way the network is assigned to him' (p. 21), rendering 'individuals' not as subject but object. The digital network architecture 'imposes its own image' on 'man' (p. 21). Our sovereignty is 'diffracted' externally through the apparatus of the digital network 'in the operational network of institutions and programs' (p. 19). While indeed the 'machine [somewhat] does what the human wants it to do', as argued by the Yes case, 'by the same token the human puts into execution only what the machine has been programmed to do' (p. 56). Humanity, integrated with (if not subordinated to) the machine

physically and cerebrally (Baudrillard, 1993, p. 58), is once again not at the centre of 'his' universe. Whereas liberal-humanism had re-centred humanity, the rise of the computing machine threatens return to the periphery—subordinate to yet another beyond human power.

Baudrillard maintains interaction via screens 'videos and telematic possibilities' (Baudrillard, 2009, p. 19) renders the external world redundant, making 'all human presences, physical or linguistic, superfluous … involution into a micro-universe, with no reason to escape anymore' (p. 19). To this, we could add the capacity for repression and surveillance enabled by ICT, wherein information about 'you' is accessible to those who might wish to know (p. 19). The integrity of the liberal subject and the significance of privacy are thus rent by the terror of total public transparency (Baudrillard, 1993, p. 122).

Perhaps worse still, our sovereignty is diffracted internally in our own mind via the screen through which we compulsively interface with ourselves (Baudrillard, 2009, p. 19). The network's capacity to give you more information about everything, including oneself, suggests it is no longer a lack of information enabling repression but its excess, since:

> you enchain him to the pure obligation of being more and more connected to himself, more and more closely connected to the screen, in restless circularity and auto-referentiality as an integrated network (Baudrillard, 2009, p. 19).

It is in relation to the compulsion to remain connected (p. 18), that Baudrillard speaks of the effect of this paradigm as being drug and coma-like (1993, p. 55); producing a subject of diminished vitality (Rubinstein, 2009, p. 162). Twenty-first century net-worked 'society' for which coding in schools aims to integrate the individual is like a never-ending hall of mirrors.

With this, the challenge to liberty emerges on two additional fronts. One is our present-day liberation from the constraints of 'traditional' existential categories enabled through new technologies making the lack of a referent 'seemingly' inconsequential (Baudrillard, 1995, p. 21; Rubinstein, 2009, p. 155; Baudrillard, 1993, p. 9). The other is liberation from meaning, which is deeply problematic, particularly in terms of liberal-humanist ideals around dispelling ignorance to fostering reasoned rational choice, and thus liberty. The problem confronting the reality principle is the increasing production of meaninglessness (Baudrillard, 1995, p. 53).

Long considered fundamental to modern liberal society, hence education, information (by which we know reality) is argued by Baudrillard (1995) to be destructive of the social; it is excess being 'directly destructive of meaning and signification' (p. 53). Excess information 'devours its own content … and the social' (p. 53). In the act of staging meaning—which is a performative anti-theatre of communication or a simulacrum of communication and meaning-making—it exhausts itself of meaning. This requires significant energies and resources to hold off 'the obvious reality of the loss of meaning' (p. 54). In making commu-nication seem more real than real (like reality TV) - because the real is no longer extant - what is produced is not reality, but hyper-reality (p. 53). Excess information produced for, about and by otherwise separated networked individuals does not invigorate the social as computational thinking and participation propose, nor does abstracted participation in such. Rather it supports the emergence of the masses and increasing meaninglessness.

Excess information, produced in hyper-reality without analogue referents, results in the 'brutal loss of signification in every domain' (p. 52), leading to escalating efforts to reinject meaning into both message and content (like reality TV), which fails because meaning is lost faster than it can be reinjected (p. 52). With this, the masses emerge as 'the aggregate left in place by the operations of the code', or model (abstract representation) (Robinson, 2012). They are 'what remains when the social has been completely removed' (Baudrillard, 1983, pp.6–7). Arguably, as noted, so-called fake news and its viral dispersal comprise but one recent expression of this (Dockray, 2017; Meyer, 2018), in relation to which the system scrambles to prevent its logic contaminating the credibility of information to production of the real and the social.

Moreover, the response of the 'system' to this logic is significant in relation to compulsory coding in schools. In this third order, a system intolerant of the silence of the masses is compelled to make them speak. It responds by reinjecting yet more information in an attempt to recover the social and the real. This is achieved through the endless construction of crisis, which the system will only too happily provide a 'solution' to (securing its own validity); a staging of death as a basis for resurrection (Duarte, 2016, p. 471; Baudrillard, 1994, p. 19). The system also responds through doubling down on the same quantic logic which Baudrillard holds responsible for the production of unreality and the masses. Endless polls, surveys, tests (Baudrillard, 1983, p. 20), big data and predictive analytics are all designed to 'capture' and reproduce reality, for which we need to produce even more coders—but which instead possibly produces even more abstraction and thus simply a simulacrum of the social. This paradox is surely captured in Puiu's statement that, 'Swedish kids will learn programming from their first year in primary school. They'll also learn how to spot fake news' (Puiu, 2017, para 1).

Conclusion

> The extent of knowledge is growing at such an alarming rate that very soon the 'basic need' of every normal kid will only be met by implanting computer chips on their brains (Debating Europe, 2015, para 75).

Discussion above explored the public debate around compulsory coding in schools. While considerable literature investigates how to best teach programming, which languages and technologies to introduce first and whether learning to program is inherently more difficult than other subjects, there is arguably little formal literature that has explored making coding compulsory in schools. Moreover, there has been little to no formal investigation of the public response to this coding in school initiative, a significant absence. Not because what the public thinks about state education initiatives is always of research importance, but for the reason outlined by Kissinger in his discussion of the Enlightenment and artificial intelligence (2018).

Kissinger is perhaps right regarding our times. Ours is an increasingly digitally inscribed world of abstraction as acknowledged by both Yes and No cases. The difference between the cases, however, concerns the implications of digital ubiquity and education's response. Advocating compulsory coding and supported by ICT 'luminaries', the corporate sector and the state, the Yes case maintains ubiquity necessitates

compulsory coding for individual and societal well-being, or in liberal-humanist parlance, securing the kinds of knowledge allowing for the individual's exercise of rational choice. It is a response grounded in positive liberty. Framed by a negative conceptualization of liberty, the No case accords digital ubiquity less relevance in terms of justifying compulsion. Discourse deployed by both constructs the truth of their positions on the grounds of negating disadvantage. Here too argument is framed by these two different conceptualizations of liberty. The Yes case supports a Rawlsian distributive model, while the No case—though somewhat libertarian—asserts that market intervention disadvantages those who acquired existing (coding) resources and remunerations through hard work and talent.

It is the habits of mind discourse, however, which 'seemingly' places the Yes case outside of the otherwise (relatively) consensual liberal-humanist framing of the ubiquity and disadvantage discourses. This, I have argued, is the most significant—though arguably 'qualified' point of difference between the cases. While the Yes case's emphasis on computational thinking and participation, critical thinking and the cognitive and cerebral seems to be at odds with the human-focused goals of liberal-humanism, otherwise captured by the No case and its analogue referents, to state that this view places the Yes case beyond the boundaries of liberal-humanism is to over-stretch the argument. Such conclusion would ignore liberal-humanism's equal focus upon the science of 'Humankind *and* Nature'. Kissinger's discussion acknowledges that it was Enlightenment philosophy and 'values' that facilitated the 'rise of the machine'.

The Enlightenment we are told, 'started with essentially philosophical insights spread by a new technology' (Kissinger 2018, para 29), but with the rise of the machine, with which school children are compelled to interface, 'our period is moving in the opposite direction. It has generated a potentially dominating technology in search of a guiding philosophy' (para, 29). Hence, discussion above references Baudrillard's argument around deepening abstraction, not to invalidate the Yes case, but to suggest how its implications might damage key liberal-humanist tenants, which it otherwise upholds. In other words, Baudrillard offers a deeply sociological account of the logics of our times. And like Kissinger, Baudrillard sees these logics as deeply problematic for a number of reasons, not the least of which is their capacity to deconstruct notions of self, society and liberty as conceived by 'traditional' liberal-humanism.

For Baudrillard, it is the deepening engagement with abstraction, which is potentially most problematic. Physical and cerebral integration with ICTs blurs boundaries between subject and object, while excess information results in a loss of the social and meaninglessness—outcomes fundamentally opposed to key tenants of liberal humanism and the achievement of liberty. These are not insignificant outcomes. If Baudrillard is correct—and time will tell—then it would seem timely for the debate around compulsory coding in schools to go beyond the assumption that those who oppose this initiative are simply obstructionist Luddites who entertain a romantic view of children grounded in a bygone era. Arguably, the debate also needs to go beyond the sometimes simplistic response to children using computers, that too much screen time is damaging (Etchells, 2019). Rather, what this analysis of the popular response to compulsory coding in schools has shown is that there is

possibly more at stake than media effects (DiPietro, Ferdig, Boyer, & Black, 2007). Increased human enmeshment in the digital 'network', which is now compulsory, raises more significant issues. Hence, the challenge arising from the compulsion to code is arguably that signalled by Kissinger. With humanity's movement along a trajectory of deepening engagement with the digital, which does not show any signs of reversing soon, perhaps, we are indeed in need of a new guiding philosophy (2018).

Abbreviations
CEO: Chief executive officer; ICT: Information communication technologies; STEM: Science, technology, engineering, mathematics

Authors' contributions
LT, who is the sole author, carried out the research project from conceptualisaton to writing up of findings. The author read and approved the final manuscript.

Availability of data and materials
All data sources identified in this paper are in the public domain and can be accessed online.

Competing interests
The author declares that he has no competing interests.

References
Abdul-Rahman, S., & Bendeict, B. (2014). Learning programming via worked-examples: Relation of learning styles to cognitive load. *Computers in Human Behavior, 20*, 286–298.
Agar, J. (2012). Raging against God: Examining the radical secularism and humanism of 'New Atheism'. *Journal of Critical Realism, 11*(2), 225–246.
Altheide, D. (2003). Notes towards a politics of fear. *Journal for Crime, Conflict and the Media, 1*, 37–54.
AnonymousCoward. (2012). Compulsory coding in schools: The new nerd tourism. Retrieved from: https://forums.theregister.co.uk/forum/1/2012/04/18/bbc_nerd_tourism/#c_1382768
Atwood, J. (2012). Please don't learn to code. [Web log post]. Retrieved from Retrieved from Coding Horror (2017, December 15). Retrieve from: https://blog.codinghorror.com/please-dont-learn-to-code/
Atwood, J. (2015). Jeff Atwood: Learning to code is overrated. Retrieved from: http://www.nydailynews.com/opinion/jeff-atwood-learning-code-overrated-article-1.2374772
Bagshaw, E. (2015). Tony Abbott ridicules his own party in school coding gaffe. Retrieved from: http://www.smh.com.au/national/education/tony-abbott-ridicules-his-own-party-in-school-coding-gaffe-20150528-ghbdal.html
Baine, G., & Barnes, I. (2014). Why is programming so hard to learn? In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 356–356). Uppsala: ITICSE '14, June 21–25, 2014.
Bajarin, T. (2014). Why basic coding should be a mandatory class in Junior High. Retrieved from: http://time.com/2881453/programming-in-schools/
Baudrillard, J. (1983). In the shadow of the silent majorities--or the end of the social, and other essays. Semiotext.
Baudrillard, J. (1993). *The transparency of evil: Essays on extreme phenomenon*. London: Verso.
Baudrillard, J. (1994). Simulacra and Simulation. Ann Arbor. University of Michigan Press.
Baudrillard, J. (1995). Simulacra and simulation. In *Body, in theory: Histories of cultural materialism*. University of Michigan Press.
Baudrillard, J. (2009). The vanishing point of communication. In D. Clarke, M. Doel, W. Merrin, & R. Smith (Eds.), *Jean Baudrillard: Fatal theories* (pp. 16–23). New York: Routledge.
Bethune, J. (2016). Classroom coding: Programming makes its way into Japanese schools. Retrieved from: https://metropolisjapan.com/classroom-coding/
Boston Dynamics. (2017). Retrieved from: https://www.bostondynamics.com/
Bresnihan, N., Millwood, R., Oldham, E., Strong, G., & Wilson, D. (2015). A critique of the current trend to implement computing in schools. *Pedagogika, 65*(3), 292–300.
Broadband Solutions. (2016 The importance of coding in education. Retrieved from: https://www.broadbandsolutions.com.au/business-centre/viewpoint/importance-of-coding-in-education
Burke, Q, O'Bryne, W. & Kafai, Y. (2016). Computational participation: Understanding coding as an extension of literacy instruction.
Burks, D. (2017). Coding as the new literacy why everyone should learn to code! HTML [500]. [Web log post]. Retrieved from Burks (2017, December 15). Retrieved from: http://www.thehtml500.com/en/blogs/coding-as-the-new-literacy
Byrne, C. (2013). No–you don't need to learn to code. Retrieved from: https://www.fastcompany.com/3020126/no-you-dont-need-to-learn-to-code

Calixto, J. (2015). Aussie kids learning computer coding at primary schools. from: https://www.sbs.com.au/news/aussie-kids-learning-computer-coding-at-primary-schools

Campbell, C., & Walsh, C. (2017). Introducing the 'new' digital literacy of coding in the early years. *Practical Literacy, 22*(3), 10–12.

Cardell-Oliver, R. (2014). How students experience learning to program. *Educational and Research Perspectives, 41*, 196–216.

Carroll, N. (2016). "Coding in every Australian school". Accessed 15 Dec 2017. Retrieved from: https://d3n8a8pro7vhmx.cloudfront.net/australianlaborparty/pages/2907/attachments/original/1444100065/Coding_in_Schools.pdf?1444100065

Cartelli, A. (2002). Computer science education in Italy: A survey. *SIGCSE Bulletin, 34*(4), 36–39.

Cellan-Jones, R. (2014). Are teachers ready for the coding revolution? Retrieved from: http://www.bbc.co.uk/news/technology-25857276

Chambers, S. (2014). Why schools in England are teaching 5-year-olds how to code. Retrieved from: https://www.bloomberg.com/news/2014-10-15/why-schools-in-england-are-teaching-5-year-olds-how-to-code.html

Chris-Granger.com. (2015 Coding is NOT the new literacy. Retrieved from: http://www.chris-granger.com/2015/01/26/coding-is-not-the-new-literacy/

Clark, Bryan. (2018 Ex-Googler warns coding bootcamps are lacking in two key areas. Retrieved from: https://thenextweb.com/dd/2018/02/09/ex-googler-warns-coding-bootcamps-are-lacking-in-two-key-areas/

Clarke, J. (2013). Rise of the machines: How computers took over the stock market. Retrieved from: http://www.theregister.co.uk/2013/06/21/hft_financial_trading_rise_of_the_machines/

Cohen, S. (2015). It's time we make 'code' an official language and teach it in every school. Retrieved from: https://www.digitaltrends.com/computing/its-time-we-made-code-an-official-language-teach-it-in-school/

Coleman, J. (1976). Rawls, Nozick, and educational equality. *The Public Interest, 43*, 121–128.

Computing Education Research Blog. (2014). ISTE debate: Should we teach computer science in elementary school? [Web log post]. Retrieved from Computing Education Research Blog (2017, 16 December). Retrieved from: https://computinged.wordpress.com/2014/12/08/iste-debate-should-we-teach-computer-science-in-elementary-school/

Crittenden, B. (2006). The school curriculum and liberal education. *Research and Perspectives, 33*(1), 105–127.

Crow, D. (2014). Why every child should learn to code. Retrieved from: https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick

Deagon, A. (2018). Liberal secularism and religious freedom in the public space: Reforming political discourse. *Harvard journal of law & public policy, 41*(3), 901–934.

Debate.org. (2013). Should programming be mandatory in elementary school? Retrieved from: http://www.debate.org/debates/Should-Programming-be-mandatory-in-Elementary-School/1/

Debating Europe. (2015). Should computer science be taught in elementary schools in Europe? Retrieved from: http://www.debatingeurope.eu/2015/01/22/should-computer-science-be-taught-in-elementary-schools-in-europe/#.WhzooEqWZQJ

Della Cava, M. (2015). Should students learn coding? Students, schools disagree, poll Finds. Retrieved from: https://www.usatoday.com/story/tech/2015/08/20/google-gallup-poll-finds-parents-want-computer-science-education-but-administrators-arent-sure/31991889/

Deloitte Access Economics. (2016). Australia's digital pulse: Developing the digital workforce to drive growth in the future. Retrieved from: https://www.acs.org.au/content/dam/acs/acs-documents/PJ52569-Australias-Digital-Pulse-2016_LAYOUT_Final_Web.pdf

Dimeo, J. (2017). Re-coding literacy. Retrieved from: https://www.insidehighered.com/digital-learning/article/2017/09/06/professor-writes-everyone-should-code-movement-re-coding

DiPietro, M., Ferdig, R., Boyer, J., & Black, E. (2007). Towards a framework for understanding electronic educational gaming. *Journal of Educational Multimedia and Hypermedia, 16*(3), 225–248.

Dockray, S. (2017). Fake news, artificial intelligence and data visualisation. *Artlink, 37*(1), 12–17.

Dodge, D. (2017). Is computer coding for kids a fad or an essential new literacy? Retrieved from: https://codakid.com/is-computer-coding-for-kids-a-fad-or-an-essential-new-literacy/

Duarte, M. (2016). Educating citizens for humanism: Nussbaum and the education crisis. *Studies in Philosophy and Education, 35*(5), 463–476.

Dvorak, J. (2014). Teaching coding to kids is a scam. Retrieved from: https://www.nytimes.com/roomfordebate/2014/05/12/teaching-code-in-the-classroom/teaching-coding-to-kids-is-a-scam

ECDL. (2015). Computing and digital literacy call for a holistic approach. Retrieved from: http://ecdl.org/media/position_paper_-_computing_and_digital_literacy.pdf

Editorial Board. (2017). Coding should be available in classrooms, not compulsory. http://www.queensjournal.ca/story/2017-07-30/editorials/coding-should-be-available-in-classrooms-not-compulsory/

Estrin, D. (2017). In Israel, teaching kids cyber skills is a national mission. Retrieved from: https://www.usnews.com/news/world/articles/2017-02-04/in-israel-teaching-kids-cyber-skills-is-a-national-mission

Etchells, P. (2019). Five damaging myths about video games – let's shoot 'em up. The Guardian, 7th April. Retrieved from: https://www.theguardian.com/games/2019/apr/06/five-damaging-myths-about-video-games-lets-shoot-em-up

Euractive. (2015). Infographic: Coding at school — How do EU countries compare? Retrieved from: https://www.euractiv.com/section/digital/infographic/infographic-coding-at-school-how-do-eu-countries-compare/

European Commission. (2016). Coding - the 21st century skill. Retrieved from: https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill

Evlin, L. (2017). Australia must embrace AI revolution with automation set to affect Every Job, Report Says. Retrieved from: http://www.abc.net.au/news/2017-08-08/australia-must-embrace-ai-revolution-alphabeta-report/8774044

FamilyTech (2017). The one skill your kid needs to excel in life: Coding. Retrieved from: http://familytech.com/the-one-skill-your-kid-needs-to-excel-in-life-coding/

Felker, C. (2013). Maybe not everybody should learn to code. Retrieved from: https://slate.com/technology/2013/08/everybody-does-not-need-to-learn-to-code.html

Fernandez, S. (2008). *Teaching and learning languages other than English (LOTE) in Victorian schools*. Victoria: Department of Education and Early Childhood Development Issue 14. Retrieved from: https://www.education.vic.gov.au/Documents/school/teachers/teachingresources/discipline/languages/langlearnreport.pdf.

Fields, D., Giang, M., & Kafai, Y. (2014). *Programming in the wild: Trends in youth computational participation in the online scratch community* (pp. 2–11). Berlin: WiPSCE 2014 Proceedings.

Foucault, M. (1972). *The archaeology of knowledge and the discourse on language*. Translated by Alan Sheridan Smith. New York: Pantheon Books.

Foucault, M. (1981). The order of discourse. In R. Young (Ed.), *Untying the text: A post-structuralist reader* (pp. 48–78). Boston: Routledge & Kegan Paul.

Gahringer, R. (1995). Liberalism and humanism. *Ethics, 66*(1), 36–50.

Geggel, L. (2018). Why does the Tesla look so fake in space? We asked a chemist. Retrieved from: https://www.livescience.com/61690-why-spacex-roadster-looks-fake.html

Gilbert, P. (2017). Coding should be SA's 12th language. Retrieved from: http://www.itweb.co.za/index.php?option=com_content&view=article&id=166546:Coding-should-be-SA-s-12th-language&catid=660

Good, G. (2001). *Humanism betrayed: Theory, ideology, and culture in the contemporary university*. McGill-Queen's University Press.

Gouvea, R., & Vora, G. (2018). Creative industries and economic growth: Stability of creative products exports earnings. *Creative Industries Journal, 11*(1), 22–53.

Gover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Research, 42*(1), 38–43.

Gray, P. (2014). Not everyone needs to learn to code. Retrieved from: https://www.techrepublic.com/article/not-everyone-needs-to-learn-to-code/

Guilherme, A. (2016). Do we have a right to education or a duty to educate ourselves? An enquiry based on Fichte's views on education. *Power and Education, 8*(1), 3–16.

Guzdial, M. (2014). The danger of requiring computer science in K-12 schools. [web log post]. Retrieved from Communications of the ACM (2016, December 16). Retrieved from: https://cacm.acm.org/blogs/blog-cacm/173870-the-danger-of-requiring-computer-science-in-k-12-schools/fulltext

Guzdial, M., Hohmann, L., Konneman, M., Walton, C., & Soloway, E. (1998). Supporting programming and learning-to- program with an integrated CAD and scaffolding workbench. *Interactive Learning Environments, 6*(1–2), 143–179.

Haaramo, E. (2015). Building a digital future: Should coding be mandatory for every schoolchild? Retrieved from: http://www.zdnet.com/article/building-a-digital-future-should-coding-be-mandatory-for-every-schoolchild/

Hait, B. (2017). Ellenbrook Primary School asks Education Minister to exempt 'disadvantaged' kids from compulsory foreign language classes. Retrieved from: https://thewest.com.au/news/education/ellenbrook-primary-school-asks-education-minister-to-exempt-disadvantaged-kids-from-compulsory-foreign-language-classes-ng-b88661297z

Hall, Z. (2017). Tim Cook believes learning to code is more important than English. Retrieved from: https://9to5mac.com/2017/10/10/tim-cook-interview-video/

Hamilton-Smith, L. (2017). Learning curve: Coding classes to become mandatory in Queensland schools. Retrieved from: http://www.abc.net.au/news/2016-11-17/coding-classes-in-queensland-schools-mandatory-from-2017/8018178

Hiltunen, T. (2016). *Learning and teaching programming skills in Finnish Primary Schools – The potential of games*. Masters Thesis. University of Oulu Department of Information Processing.

Hinsliff, G. (2015). Should kids learn to code? Retrieved from: https://www.theguardian.com/news/2015/dec/03/should-kids-learn-code

Howard, C. (2013). F-35 Joint Strike Fighter benefits from modern software testing, quality assurance. Retrieved from: http://www.militaryaerospace.com/articles/2013/10/software-code-f-35.html

Hubwieser, P. (2012). Computer science education in secondary schools – The introduction of a new compulsory subject. *ACM Transactions on Computing Education, 12*(4): Article 16), 1–40.

Hutchison, A., Nadolny, L., & Estapa, A. (2015). Using coding apps to support literacy instruction and develop coding literacy. *Reading Teacher, 69*(5), 493–503.

Joinson, A. (2005). Internet behaviour and virtual methods. In C. Hine (Ed.), *Virtual Methods: Issues in social research on the internet* (pp. 21–34). New York: Berg.

Jovanov, M., Stankov, E., Mihova, M., Ristov, S., & Gusev, J. (2016). *Computing as a new compulsory subject in the Macedonian primary schools curriculum* (pp. 680–685). Abu Dhabi: IEEE global engineering education conference, 10–13 April.

Kafai, Y., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press.

Kafai, Y., & Burke, Q. (2013). Computer programming goes back to school. *Kappanmagazine.org, 95*(1), 61–65.

Kainulainen, P. (2013). August 24. In *The way of the hipster* Retrieved from: https://www.petrikainulainen.net/software-development/technology-evaluation/the-way-of-the-hipster/.

Kalelioğlu, F., & Yasemen, G. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33–50.

Keenihan, S. (2017). Revealed today, Elon Musk's new space vision took us from Earth to Mars, and back home again. In *The conversation* Retrieved from: http://theconversation.com/revealed-today-elon-musks-new-space-vision-took-us-from-earth-to-mars-and-back-home-again-84837.

Keneally, P. (2015). Let's pause before drinking the 'coding in schools' Kool-Aid. Retrieved from: https://www.theguardian.com/commentisfree/2015/jun/03/lets-pause-before-drinking-the-coding-in-schools-kool-aid

Kircher, T. (2015). Renaissance humanism and its discontents. *The European Legacy, 20*(5), 435–449.

Kissinger, H. (2018). How the Enlightenment ends. Retrieved from: https://www.theatlantic.com/magazine/archive/2018/06/henry-kissinger-ai-could-mean-the-end-of-human-history/559124/

Knobelsdorf, M., Magenheim, J., Brinda, T., Engbring, D., Humbert, L., Pasternak, A., Schroeder, U., Thomas, M., & Vahrenhold, J. (2015). Computer science education in North-Rhine Westphalia, Germany—A case study. *ACM Transactions on Computing Education, 15*(2), Article 9), 1–20.

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*(2), 154–171.

Leary, K. (2017). An AI that makes fake videos may facilitate the end of reality as we know it. Retrieved from: https://futurism.com/ai-makes-fake-videos-facilitate-end-reality-know-it/

Lucisano, F. (2017). Digital literacy and coding. Retrieved from: https://www.scopeiteducation.com.au/digital-literacy-and-coding

Lui, S. (2015). Australian schools to ditch History and Geography for coding. Retrieved from: https://www.lifehacker.com.au/2015/09/schools-to-ditch-history-and-geography-for-coding/

Luke, A. (1997). The material effects of the word: Apologies, 'Stolen Children' and public discourse. *Discourse, 18*(3), 343–368.

Lynch, M. (2017). Why coding should be a compulsory subject for students. Retrieved from: http://www.theedadvocate.org/why-coding-should-be-a-compulsory-subject-for-students/

Lyonnais, S. (2015). Should learning how to code be mandatory in the U.S? [Web log post]. Retrieved from Lyonnais (2017, December 16). Retrieved from: https://blogs.adobe.com/creativecloud/should-learning-how-to-code-be-mandatory-in-the-u-s/

Mathrani, A., & Agate, P. S. (2016). PlayIT: Game-based learning approach for teaching programming concepts. *Educational Technology & Society, 19*(2), 5–17.

Matousek, M. (2018). Elon Musk is launching his Tesla Roadster into space to prove a point — But it's also a brilliant marketing move. Retrieved from: https://sports.yahoo.com/elon-musk-launching-tesla-roadster-172511191.html

Mayo, L. (2013). Coding crisis: Getting tech skills taught in schools. Retrieved from: http://www.abc.net.au/technology/articles/2013/10/08/3864305.htm

McAllister, N. (2008). Should computer programming be mandatory for U.S. students? Retrieved from: https://www.infoworld.com/article/2633890/application-development/should-computer-programming-be-mandatory-for-u-s--students-.html

Merkouris, A., Chorianopoulo, K., & Kameas, A. (2017). Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education, 17*(2), 1–22.

Meyer, R. (2018). March 8. In *The grim conclusions of the largest-ever study of fake news* Retrieved from: https://www.theatlantic.com/technology/archive/2018/03/largest-study-ever-fake-news-mit-twitter/555104/.

Mouritsen, P., & Olsen, T. (2013). Liberalism and the diminishing space of tolerance. In J. Dobbernack & T. Modood (Eds.), *Tolerance, intolerance and respect* (pp. 127–157). London: Palgrave Macmillan.

Northrop Grumman. (2015). X-47B UCAS makes aviation history … again!: Successfully completes first ever autonomous aerial refuelling demonstration. Retrieved from: http://www.northropgrumman.com/Capabilities/x47bucas/Pages/default.aspx 2014.

Pellas, N., & Efstratios, P. (2017). Leveraging Scratch4SL and Second Life to motivate high school students' participation in introductory programming courses: Findings from a case study. *New Review of Hypermedia and Multimedia, 23*(1), 51–79.

Phillips, Jess & Bartlett, J. (2018). Should anonymous social media accounts be banned? *The Guardian*, 30 September 2018. Retrieved from: https://www.theguardian.com/media/2018/sep/30/social-media-anonymity-ban-debate-trolls-abuse%2D%2Djess-phillips-jamie-bartlett

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education, 128*, 365–376.

Pritchard, S. (2016). September 13. In *Hipsters and artists are the gentrifying foot soldiers of capitalism* Retrieved from: https://www.theguardian.com/commentisfree/2016/sep/13/hipsters-artists-gentrifying-capitalism.

Puiu, T. (2017). Swedish kids will learn programming from their first year in primary school. In *They'll also learn how to spot fake news* Retrieved from: https://www.zmescience.com/science/news-science/sweden-programming-fake-news/.

Queensland Government. (2015). #codingcounts: A discussion paper on coding and robotics in Queensland schools. Retrieved from: http://advancingeducation.qld.gov.au/SiteCollectionDocuments/Coding-and-robotics-booklet.pdf

Raja, T. (2014). Is coding the new literacy? Retrieved from: http://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education

Rawlins, L. (2015). Compulsory coding in school debate continues. Retrieved from: http://www.itweb.co.za/index.php?option=com_content&view=article&id=147080

Rawlinson, K. (2015). Microsoft's Bill Gates insists AI is a threat. Retrieved from: http://www.bbc.com/news/31047780

Rawls, J. (1999). *A theory of justice.* Oxford: OUP.

Repp, K. (2000). More corporeal, more concrete: Liberal humanism, eugenics, and German progressives at the last Fin de Siècle. *The Journal of Modern History, 72*(3), 683–730.

Riberio, V. (2014). What principle of justice for basic education? *Cadernos de Pesquisa, 44*(154), 1095–1109.

Robinson, A. (2012). Jean Baudrillard: The masses. Retrieved from: https://ceasefiremagazine.co.uk/in-theory-baudrillard-12/

Rogers, C. (2018). Falcon Heavy launch leaves inspiration in its wake. Retrieved from: https://medium.com/@rogerstigers/falcon-heavy-launch-leaves-inspiration-in-its-wake-d186a1bdb8e7

Rubinstein, D. (2009). Reality: now and then. In D. Clarke, M. Doel, W. Merrin, & R. Smith (Eds.), *Jean Baudrillard: Fatal Theories* (pp. 147–162). New York: Routledge.

Ruseva, G , Rissola, A. (2016). The role of non-formal education in teaching coding. Position paper Telecentre Europe and Public Libraries 2020. Retrieved from: https://joinup.ec.europa.eu/sites/default/files/document/2017-03/coding_policy_paper-final.pdf

Rwodzi, C. (2014). Liberal humanism in a transforming Post-Apartheid curriculum of South Africa: An introspection. *Mediterranean Journal of Social Sciences, 5*(20), 1916–1920.

Saez-Lopez, J., Gonzalez, M., & Vazquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools. *Computers & Education, 97*, 129–141.

Saines, G. (2014). 3 Reasons why "computational literacy" is ruining coding education. [Web log post]. Retrieved from Saines (2017, December 17). Retrieved from : http://blog.codecombat.com/3-reasons-why-computational-literacy-is-ruining-coding-education/

Schatzki, T. (1995). Objectivity and rationality. In W. Natter, T. Schatzki, & J. Jones (Eds.), *Objectivity and its other* (pp. 137–160). New York: Guilford Press.

Schroeder, J. (2018). Toward a discursive marketplace of ideas: Reimaging the marketplace metaphor in the era of social media, fake news, and artificial intelligence. *First Amendment Studies, 51*(1–2), 38–60.

Shein, E. (2014). Should everybody learn to code? *Communications of the ACM, 57*(2), 16–18.

Sherridan, A. (1980). *Michel Foucault: The will to truth.* London: Routledge. Tavistock Publications.

Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158.

Singh, B. (1998). Liberalism, parental rights, pupils' autonomy and education. *Educational Studies, 24*(2), 165–182.

Smith, M. (2016). Computer science for all. In *The Whitehouse Briefing Room* 30th January. [web log post]. Retrieved from The Whitehouse (2017, December 17). Retrieved from: https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all.

Sololearn. (2017). Why not to teach programming languages in all schools? Retrieved from: https://www.sololearn.com/Discuss/394837/why-not-to-teach-programming-languages-on-all-schools/

Stucky, B. (2015). Teaching coding in schools: Absolutely necessary or another fad to waste teachers time? Retrieved from: https://www.aare.edu.au/blog/?p=1076

Tamatea, L. (2005). The Dakar framework: Constructing and deconstructing the global neo-liberal matrix. *Globalisation, Society and Education, 3*(3), 311–334.

Tamatea, L. (2008). George Bush's No Child Left Behind education policy: War, ambivalence, and mimicry — online. *Review of Education, Pedagogy & Cultural Studies, 30*(2), 115–139.

Tamatea, L. (2011). Ajeg Bali discourse: Globalisation, fear and othering – online. *Asian Ethnicity, 12*(2), 155–177.

Tamatea, L., & Paramatasari, A. (2018). Bourdieu and programming classes for the disadvantaged: A review of current practice as reported online — Implications for non-formal coding classes in Bali. *Research and Practice in Technology Enhanced Learning, 12*(1), 1–27.

Tarnoff, B. (2017). September 22. In *Tech's push to teach coding isn't about kids' success – It's about cutting wages* Retrieved from: https://www.theguardian.com/technology/2017/sep/21/coding-education-teaching-silicon-valley-wages.

Team ISTE. (2015). Hotly contested: Should coding be mandatory? Retrieved from: https://www.iste.org/explore/articleDetail?articleid=470&.

Thoreau, M. (2017). Code literacy: Increasing K-12 coding education in Ohio. Retrieved from: https://ohioline.osu.edu/factsheet/cdfs-1574

Titlestadt, P. (2010). Liberalism. *English Academy Review, 27*(2), 94–100.

Turula, T. (2017). Sweden's government is about to introduce coding in schools — From first grade. Retrieved from: http://nordic.businessinsider.com/swedens-government-gives-full-support-behind-coding-from-grade-1-2017-3/

Ungar, S. (2001). Moral panic versus the risk society: The implications of the changing sites of social anxiety. *British Journal of Sociology, 3*(2), 271–292.

Van Zyl, S., Elsa Mentz, D., & Havenga, M. (2016). Lessons learned from teaching Scratch as an introduction to object-oriented programming in Delphi. *African Journal of Research in Mathematics, Science and Technology Education, 20*(2), 131–141.

Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies, 1*(2), 42–64.

Verma, A. (2016). Japan just made computer programming a compulsory subject in its schools. Retrieved from: https://fossbytes.com/japan-computer-programming-compulsory-subject-schools/

Vincent, J. (2017a). Did Elon Musk's AI champ destroy humans at video games? It's complicated. Retrieved from: https://www.theverge.com/2017/8/14/16143392/dota-ai-openai-bot-win-elon-musk

Vincent, J. (2017b). All of these faces are fake celebrities spawned by AI. Retrieved from: https://www.theverge.com/2017/10/30/16569402/ai-generate-fake-faces-celebs-nvidia-gan

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715–728.

Warner Brothers. (1999). The matrix.

West, E. (1965). Liberty and education: John Stuart Mill's dilemma. *Liberty, 40*(52), 129–142.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society, 366*(1881), 3717–3725.

Wing, J. (2010). Research notebook - computational thinking: What and why? In *The link* Retrieved from: http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

Wing, J. (2016). Progress in computational thinking and expanding the HPC community. *Communications of the ACM, 59*(7), 10–11.

Wong, G., Cheung, H., Ching, E., & Huen, J. (2015). *School perceptions of coding education in K-12: A large scale quantitative study to inform innovative practices* (*IEEE* international conference on Teaching, Assessment, and Learning for Engineering (TALE)). Zhuhai: United International College.

Woolf, C. (2014). Reading, Math and ... Javascript? Coding is now mandatory in English schools. Retrieved from: https://www.pri.org/stories/2014-09-25/reading-math-and-javascript-coding-now-mandatory-english-schools

Young, T. (2012). Compulsory coding in schools: The new nerd tourism. Retrieved from: https://forums.theregister.co.uk/forum/1/2012/04/18/bbc_nerd_tourism/#c_1382768

Yousefi, Z., Yousefy, A., & Keshtiaray, N. (2015). Liberal humanism and its effect on the various contemporary educational approaches. *International Education Studies, 8*(3), 103–113.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.