Institute of Education, University of London

September 2012

# Computer programmers and the “bilingual advantage”: Enhanced executive control in non-linguistic interference tasks.

**Hannah Wright**

A dissertation submitted in partial fulfilment of requirements for the degree of

**MSc in Child Development**

# Acknowledgements

I am enormously grateful to all of the participants – programmers and non-programmers – who gave up their time to make this study happen. Thank you.

I am also indebted to those who helped with participant recruitment – Emma Mulqueeny, Lawrence Job and Josh Pickett at Young Rewired State; Adam Cohen-Rose; Tara Bloom; Tracey Herald; Nikki Sims; Adrian Wright and Catherine Erskine – and to the developers and computer scientists who kindly shared experiences and ideas: Paul Battley, Duncan Gough, Alex Maccaw, Dr John Murnane and Prof. Richard Noss.

Thank you to my supervisor, Prof. Andy Tolmie, for very helpful feedback throughout.

Sincere thanks to my family and friends, and particularly to Paul Sims – without your support, this would have been much, much harder.

# Abstract

Bilingualism is associated with life-long cognitive advantages. It is well established that bilinguals perform better at non-verbal tasks requiring enhanced executive control. Bilinguals typically record faster response times than their monolingual peers; this is thought to result the development of greater efficiencies in conflict monitoring network, which develop in response to the additional demands of managing two competing language systems. The present study investigates whether this "bilingual advantage" is also associated with the frequent use of computer programming languages. The performance of 10 professional computer programmers (aged 22–25) and 10 adolescent computer programmers (aged 14–17) is compared to age-matched and IQ-matched controls in two executive control tasks. In the Attention Networks Test, as predicted, programmers recorded faster global reaction times than their monolingual peers; the difference was significant. In the Stroop colour-word task, programmers recorded slower reaction times; however, these results were not significant. Overall, the results suggest that extensive computer programming experience may, like bilingualism, be associated with enhanced executive control. Whatever the direction of this relationship, it could have important implications for education; these are discussed, along with areas for future research.

# Table of Contents

# Introduction

## Overview

Plasticity – the ability to change as a result of experience – is a defining and enduring feature of the brain across the lifespan (Huttenlocher, 2002). Like any other intensively practiced skill – for example, music (Pascual-Leone, 2001), taxi-driving (Woollett & Maguire, 2011), or meditation (Brefczynski-Lewis, Lutz, Schaefer, Levinson, & Davidson, 2007) – the use of language affects the structure and function of the brain. For example, babies are born with the ability to distinguish between the sounds of all languages, but by the end of their first year of life, any redundant connections between neurons (*synapses*) have been pruned, so that their phonetic perception is strongly biased towards the sounds that make up the languages to which they have been exposed (Kuhl, 2004). Some researchers have even suggested that the use of language is “what makes us smart” (Spelke, 2003). It is certainly difficult to imagine a skill that is practiced more frequently and acquired more universally.

Bilinguals are defined as people who require and use two (or more) languages in their every day lives (Grosjean, 1992). The term applies to individuals exposed to two languages from birth, who undergo what Meisel (1989) described a *bilingual first language acquisition,* and also to those who are immersed in additional languages later in life, who must overcome interference from their more entrenched first language through some degree of neuronal reorganisation (Hernandez, Li, & MacWhinney, 2005). Globally, bilingualism is the norm, not the exception: Crystal (1997) estimates that some two thirds of the world’s population growing up in bilingual or multilingual environments.

In stark contrast to early suspicions that bilingual children were at risk of retardation or at best, “mentally confused” (Bialystok, 2005), recent research links bilingualism to *cognitive reserve* and suggests it may offer protection against dementia in old age. Cognitive reserve describes a kind of resilience which appears to mediate the relationship between brain pathology and the clinical expression of that pathology; it is thought that this resilience derives from more efficient use of brain networks and/or the ability to deploy differential brain networks to a given task (Stern, 2002). Using a sample drawn from a memory clinic, Bialystok, Craik and Freedman (2007) found that bilingual patients presented symptoms of dementia three to four years later than monolingual patients. A second study, featuring a different sample, found that the first clinical appointments of bilinguals occurred 4.3 years later and the estimated age of onset (based on self or relative reports) was 5.1 years later than monolinguals. Although these results should be viewed with some caution due to social and cultural differences between participants, the difference is dramatic and has understandably caused much excitement about the potential impact of bilingualism on the efficiency and resilience of the brain.

The suggestion that bilingualism could enhance general cognitive performance is not new. Following the first reports of positive cognitive effects in Peal and Lambert’s (1962) landmark study, bilingualism has been linked to numerous cognitive benefits including improved metalinguistic awareness (Ben-Zeev, 1977; Ianco-Worrall, 1972), creativity (Kessler & Quinn, 1987; Ricciardelli, 1992) and problem-solving (Kessler & Quinn, 1980). In recent years, an influential area of research has explored the relationship between bilingualism and executive control, that is, the ability to selectively attend to stimuli and to inhibit inappropriate responses in order to achieve desired goals. A large number of studies (reviewed below) have found that bilinguals outperform their monolingual peers in tests involving cognitive conflict. Most of this research has focused on individuals who were bilingual from birth or early childhood, but a recent study found similar advantages in late-proficiency Chinese-English bilinguals who acquired their second language between the ages of 12 and 19 years old (Tao, Marzecova, Taft, Asanowicz, & Wodniecka, 2011). It is widely hypothesized that the bilingual advantage in these tasks arises because the bilingual brain places additional demands on a domain-general aspect of executive control in order to selectively attend to two competing language systems (e.g. Bialystok, Craik, Green & Gollan, 2009; Costa, Hernandez, Costa-Faidella, & Sebastian-Galles, 2009; Hinchley & Klein, 2011).

Like bilingualism, computer programming has often been argued to convey cognitive advantages; however, the results to date are equivocal (e.g. Pea & Kurland, 1984; Palumbo, 1990; Liao & Bright, 1991). Research into the cognitive consequences of computer programming has largely focussed on the problem-solving domain (Ormerod, 1990); in contrast, the present paper explores computer programmers' cognition from the perspective of language acquisition, an approach recommended by Murnane (1993; 2006). Like bilinguals, expert computer programmers successfully manage two or more separate lexicons, grammars and divergent concepts, avoiding inadvertent transfer between the two. Numerous studies of novice programmers indicate that they struggle to do achieve this division; transfer from natural language creates bugs (e.g. Soloway and Spohrer, 1989; Witschital, 1995). The present study therefore considers whether the "bilingual advantage" in executive control is found in computer programmers.

Section I explores the evidence linking bilingualism to enhanced executive control. Empirical support for two reported advantages in conflict tasks is considered: first, faster global reaction times; second, reduced interference in incongruent conditions. Two theories relating these advantages are considered: the bilingual inhibitory control hypothesis and conflict monitoring theory. Finally, a recent study indicating that the bilingual advantage is also found in late proficiency bilinguals is presented. Section II argues that the possibility that computer programming languages could convey similar cognitive benefits to bilingualism merits empirical investigation. It first considers the similarities and differences between natural languages and computer programming languages. It then presents evidence from the study of novice programmers, which indicates that successfully preventing transfer between natural language and programming languages is crucial to the development of programming expertise. Section III introduces the present study, which investigates the performance of adolescent and young adult computer programmers in two executive control tasks associated with the bilingual advantage.

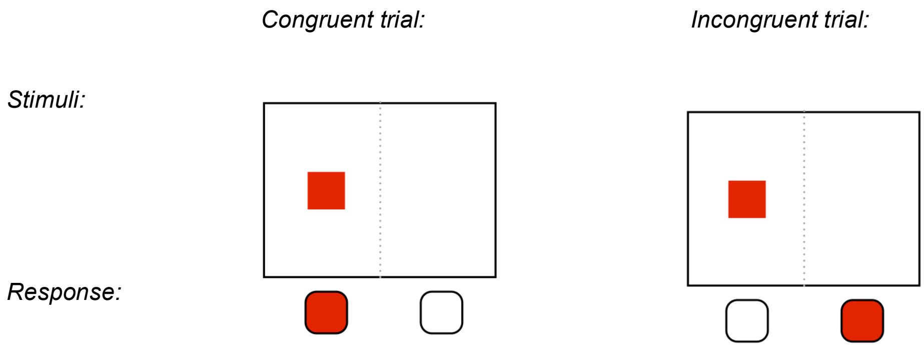
## Part I: Executive control and bilingualism

When communicating, bilinguals must successfully manage two conflicting languages; one must be accessed whilst the other is suppressed, in order to avoid involuntary language switching. The cognitive demands of this task are thought to be the origin of the bilingual advantage in executive control.

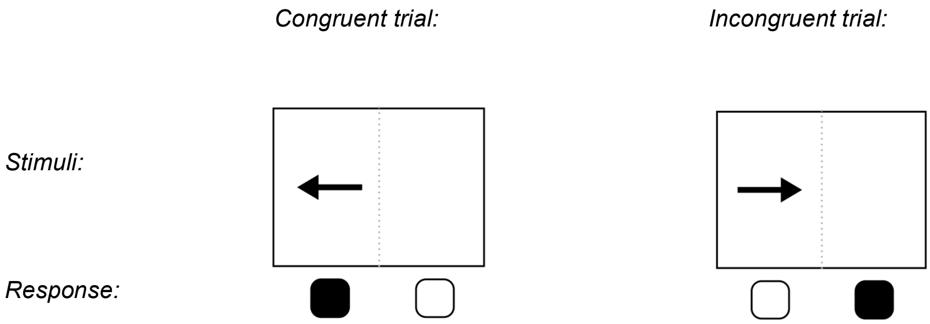
A series of studies have demonstrated that bilinguals outperform their peers on tests of non-linguistic interference. Bilingual children, middle aged adults and older adults consistently record faster global reaction times in the Simon task (Bialystok, Martin and Viswanathan, 2005; Martin-Rhee and Bialystok, 2008), the spatial Stroop/Simon arrows task (Bialystok, 2006; 2008), and flanker arrows tasks such as the Attention Networks Task (Costa, Hernandez and Sebastian-Galles, 2008; Carlson and Meltzoff, 2008; Emmorey *et al.*, 2009). These computer-based neuropsychological tasks all require participants to respond to a series of stimuli as quickly and accurately as possible; the interference comes from conditions where the stimuli and the response required are incongruent, typically resulting in slower reaction times. (See *Box 1*.)

### ***Box 1.*** **Non-linguistic interference tasks**

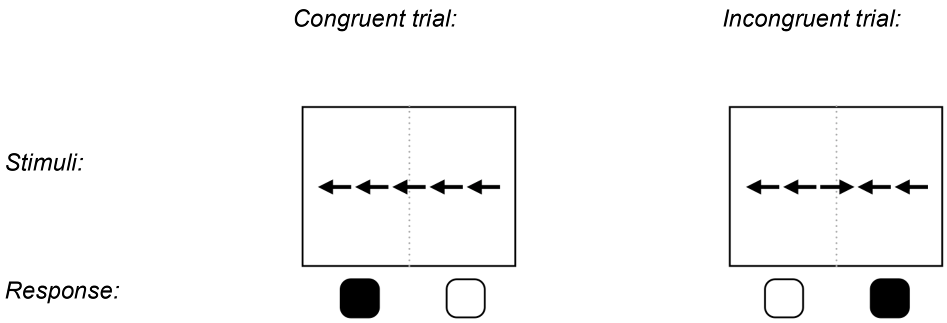
Key features of three non-linguistic interference tasks commonly used in bilingual research – the Simon task, the spatial Stroop task (sometimes referred to as a *Simon arrows task*) and the flanker task – are described here. It should be noted however different studies interpret these tasks slightly differently; this causes difficulties in directly comparing studies.



***A. Simon Task.*** The participant's task is to press one of two buttons depending on the colour of the square, as quickly and accurately as possible.  The relationship between the position of the arrow on the screen and the location of the correct response determines congruency.



***B. Spatial Stroop.*** The participant's task is to press one of two buttons depending on the direction that the arrow is pointing, as quickly and accurately as possible.  The relationship between the direction of the arrow and the location of the correct response determines congruency.



***C. Flanker Task.*** In this task, the correct response is determined by the direction in which the central arrow is pointing. Congruency is determined by the direction of the flanker arrows.

The bilingual advantage is less pronounced in young adults: studies have reported that they recorded faster global RTs in spatial Stroop (Bialystok & DePape, 2009) and flanker tasks (Costa, Hernandez, & Sebastian–Galles, 2008; Tao, Marzecova, Taft, Asanowicz, & Wodniecka, 2011); others have found that participants in this age group only perform faster than their peers in tasks involving frequent switching between congruent and incongruent trials, and not in low-switch versions of the same task (e.g. Bialystok, 2006; Costa *et al.*, 2009). A study using the Simon task with young adults found no difference between the global RTs of bilingual and monolingual participants (Bialystok, 2006). Overall, however, bilinguals tend to record faster global RTs in non-linguistic interference tests; in a recent and impressively detailed empirical review, Hilchey & Klein (2011) concluded that this finding was robust. The advantage, however, does not appear to extend to bimodal bilinguals; Emmorey *et al*. (2008) found no difference in the performance of speech-sign bilinguals and monolinguals on a flanker task.

Some studies have also reported that the bilinguals were not only faster overall but also less affected by the conflict conditions of these non-linguistic interference tests; the difference in their reaction times in congruent versus incongruent conditions was significantly smaller than for their monolingual peers. For example, Bialystok, Craik, and Luk (2008) found that the Simon effect (the increase in reaction times in incongruent conditions) was significantly smaller for older bilinguals than for age- and IQ-matched controls. The authors observed similar differences in performance on the Stroop colour-word naming task; bilinguals were less affected by the Stroop effect than their monolingual peers (Bialystok, Craik and Luk, 2008). For many years, it was this apparent advantage (rather than the global RT advantage) that attracted the greater interest from theorists; this resulted in the *bilingual inhibitory control advantage (BICA) hypothesis*. Based on Green's (1998) inhibitory control theory, the BICA hypothesis suggests that a general-purpose inhibitory control system holds additional responsibilities in the bilingual mind, enabling the speaker to ignore distractions from the irrelevant language in order to effectively communicate. It is proposed that in the bilingual brain, both languages are simultaneously activated in response to stimuli, regardless of the relative relevance of these respective languages to the speaker's context; the inhibitory control system must react to this conflict by suppressing the irrelevant information. Support for the hypothesis comes from studies indicating that parallel language activation is a feature of the bilingual brain. For example, Thierry and Wu (2007) found measured event related potentials when Chinese-English bilinguals were asked to judge the semantic relatedness of English word pairs (e.g. *train-ham*); the results suggested that participants unconsciously translated the words into Chinese in cases where the Chinese translations of the two words shared a character (e.g. the Chinese words for train and ham are ***Huo*** *Che* and ***Huo*** *Tui).* This parallel activation appears to affect on performance in comprehension and language production tasks (e.g. Phillipp & Koch, 2009).

There are two key flaws with the BICA hypothesis. First, the evidence for reduced interference effects in bilinguals is equivocal. Several researchers (e.g. Bialystok, Martin & Viswanathan, 2005; Bialystok, 2006; Colzato *et al.*, 2008) have struggled to reliably reproduce this reduced interference effect. Meanwhile, one of the most commonly cited examples of reduced Simon effect in bilinguals – a study by Bialystok, Craik, Klein, and Viswanathan (2004) – suffers from methodological flaws; the monolingual participants were all Canadian residents, whilst the bilingual participants all lived in India or Hong Kong, raising the strong possibility that social and cultural differences in the sample influenced results. Hilchey and Klein's (2011) review concludes that the bilingual advantage in conflict resolution tasks is "sporadic at best, and in some cases conspicuously absent". Further, the BICA hypothesis alone cannot explain the more reliable finding that bilinguals outperform monolinguals by similar magnitudes in both conflict and non-conflict conditions. Whilst some supporters continue to endorse the hypothesis (Philipp & Koch, 2009), others (e.g. Bialystok, 2006; Costa, 2009; Hilchey & Klein, 2011; Bialystok, Craik & Luk, in press) are increasingly identifying a domain-general c*onflict monitoring network* as a promising alternative candidate for the source of the bilingual advantage.

Botvinick and colleagues' conflict monitoring theory uses evidence from cognitive neuroscience to argue that a network of specific brain regions, particularly the dorsal anterior cingulate cortex (ACC), play a key role in detecting conflict in information processing; upon the detection of such conflicts, the network allocates additional cognitive control resources to the task in order to protect against subsequent conflict (Botvinick, Cohen, & Carter, 2004). This network has been implicated in the monitoring of conflict between two competing languages (e.g. Hernandez, Dapretto, Mazziotta, & Bookheimer, 2001) as well as more general conflict monitoring tasks. Hilchey & Klein (2011) explain the plausible theory that it is bilinguals' frequent use of this domain-general system to monitor two conflicting languages, rather than a specific inhibitory control mechanism, that is the source of the bilingual advantage in non-linguistic interference tests. Costa *et al*. (2009) provided evidence to support this theory by manipulating conflict resolution tasks with the aim of placing differential demands on the conflict monitoring network. In the high conflict-monitoring condition, the task involved rapid switching between congruent and incongruent trials; conversely, in the low conflict-monitoring condition, most of the trials were of the same type (congruent or incongruent). Bilinguals tended to perform faster in the high-monitoring condition but there was little difference between the groups in the low-monitoring condition. The authors argued that this was due to impact of bilingualism on the conflict monitoring system.

A recent neuroimaging study observed differences in the brain activity of bilinguals and monolinguals during a flanker task (Luk, Anderson, Craik, Grady, & Bialystok, 2010). The study demonstrated that both monolinguals and bilinguals showed increased activation in similar brain regions (bilateral middle occipital gyrus, left fusiform gyrus, left lingual gyrus, bilateral cerebellum, and right caudate and IFG) during successful performance on congruent tasks; however, whilst monolinguals also activated those regions during successful incongruent trial responses, bilinguals showed increased activation in different brain regions (bilateral cerebellum, bilateral superior temporal gyri, left supramarginal gyri, bilateral postcentral gyri, and bilateral precuneus). The authors interpret the findings as support for the bilingual inhibitory control hypothesis, an interpretation that appears to be logical. However, Hilchey and Klein (2011) suggest that instead, bilinguals may possess a greater ability to detect and allocate inputs to different brain regions, based on the presence or absence of conflict; this frees up valuable processing resources, thus increasing processing speed and reducing global RT. This account sits more comfortably with the evidence from behavioural studies, since the inhibitory control explanation does not explain the global RT advantage. Bialystok *et al.* (in press) suggest that the two theories need not be mutually exclusive; in fact, they argue an inhibition account is still required in addition to the conflict monitoring theory in order to explain why bilinguals sometimes show a reduced interference effect in addition to faster global reaction times.

Whatever the source of the bilingual advantage, the finding that bilinguals perform better on non-verbal conflict tasks has been widely replicated. However, the studies described above are limited in two important ways: first, they have generally focussed on early proficiency bilinguals – those who have been exposed to two languages from birth or early childhood – and therefore cannot be generalised to those who learn second languages later in life; second, they exclusively focus on balanced bilinguals – that is, bilinguals who are equally proficient in both languages – so it is not possible to discern whether or not this bilingual advantage would apply to those with one more dominant language.

Fortunately, a study from Tao *et al*. (2011) makes good progress in addressing these gaps in the literature. The authors compared the efficiency of the attention networks of young Australian adults who were English monolinguals or Chinese–English bilinguals. The bilingual group was further divided according to age of acquisition: early bilinguals, who had arrived in Australia before the age of 6, and had received their formal education in English; and late proficiency bilinguals, who had arrived in Australia between the ages of 12 and 19 years old. The efficiency of three attention networks – alerting (achieving and maintaining an alert state), orienting (selecting information from sensory input), and executive control (monitoring and resolving conflict) – as well as the hemispheric symmetry of these networks, was measured using Lateralized Attention Network Test (LANT), designed by Green *et al.* (2008). (This is an adaptation of the Attention Network Test described in detail in the methods section and *Box B* below.)

Both bilingual groups significantly outperformed the monolingual group, suggesting that the mastery of two very different grammars and lexicons produces the bilingual advantage found in bilinguals speaking two similar languages. The early proficiency group demonstrated significantly faster reaction times in all conditions; the authors suggested was due to the enhancement of monitoring systems (Costa *et al.* 2009) resulting from early bilingualism, though they also acknowledged that this could also be simply attributed to greater vigilance. Fascinatingly, however, the late proficiency bilingual showed the greatest advantage in conflict resolution conditions; the authors attributed this to greater reliance on the executive network in later second language acquisition, both in order to stave off greater interference from their “more solidified" first language and to support the processing of the weaker second language. Participants in both groups of the study were not considered balanced bilinguals; the early proficiency group were considered strongly dominant in L2, and the late proficiency group were considered moderately dominant in L1. (It should be noted that proficiency were self-rated using a 7-point Likert scale; no proficiency tests were conducted.)

In summary, this section has described evidence for a bilingual advantage in non-verbal conflict tasks. Bilingual children, middle aged and older adults perform these tasks faster than their monolingual peers; bilingual young adults also perform faster in the more cognitively demanding of these tasks. This advantage is thought to derive from a conflict monitoring network, which is enhanced in bilinguals due to its role in preventing transfer between their two languages. Most of the evidence for the global RT advantage derives from studies of balanced, early-proficiency bilinguals; however, it has also been found in late-proficiency bilinguals and those with asymmetric linguistic abilities. Some studies have also found that bilinguals demonstrate a reduced interference effect during incongruent trials; however these results have proved difficult to replicate, particularly in children and young adults. Importantly, it is generally agreed that the reason why bilinguals perform better in tests of executive control is the additional cognitive demands of managing two separate language systems.

## Section II: Computer programming and cognition

Papert's (1980) influential work raised hopes for that the Logo programming language could be used as a tool to aid the development of children's abstract thinking. To date, empirical studies have focused primarily on seeking to identify gains in general problem-solving ability as the result of programming instruction, and the evidence has been less than conclusive (Palumbo, 1990). Pea & Kurland's (1984) critical review concluded that claims of cognitive transfer could not be substantiated. Pea (1983) likened Papert's enthusiasm for the potential of Logo in education to "the overzealous prescriptions for studying Latin in Victorian times"; his implication was that Logo, like Latin, had been over-hyped and had under delivered. However, several studies have reported evidence of positive transfer. In a meta-analysis of 65 studies on the effects of computer programming on cognitive outcomes, Liao and Bright (1991) found that 89 percent reported positive results, but the overall effect size was moderate (0.41); the power of the effect could not match the power of Papert's ideas.

Both Papert's powerful ideas and Pea's criticism equate programming languages with natural languages, but to what extent is this comparison justified? In the 1970s, some American universities accepted computer programming languages in fulfilment of their foreign language requirement (Norman, 2008). Papert (1980) explicitly compares the learning of computational languages with the learning of natural languages, "one of the things children do best". More recently, Cohen and Haberman (2007) called for recognition of computer science as a high-level scientific language, "the language of technology". Weinberg (1971), however, was more cautious about the comparison: "Just calling it a language doesn't make it one."

If empirical evidence suggested that the brain utilised the natural language system to store and process computer programming languages, it would be reasonable to hypothesise than expert programmers have bilingual brains. Unfortunately the relationship between natural language processing and computer programming languages has merited little empirical attention (Murnane, 1993; 2006). Studies of programming tend to view it as a problem solving – rather than a linguistic – activity, so there has been little exploration of the relationship between the vocabulary, grammar and syntax of programming languages and programmers' semantic and conceptual knowledge (Ormerod, 1990).

A computer program is a text combining a data structure and a set of instructions that enable the computer to calculate desired functions, written according to strict grammatical rules that can be interpreted by the computer (Détienne, 2002). A programming language is a semiotic code comprising vocabulary and grammar used to convey instructions to the computer but also to other programmers, many of whom may contribute to a single piece of software. There are, of course, many differences between natural languages and programming languages. One difference that is often cited as the most important (e.g. Detienne 2002) is that while in natural languages statements may be open to interpretation, in programming languages they must be clear and unambiguous; computer programmers are strongly constrained by the syntactic rules of the programming languages.

Hockett (1960) isolated 13 features that characterize human language and which distinguish it from other communication systems. Weinberg (1971) noted several discrepancies between these and the characteristics of programming languages. Unlike spoken languages, programming languages contain no vocal-auditory channel; and because they are written rather than spoken, there is no rapid fading – instead, they must be explicitly erased. Spoken languages are characterized by *broadcast transmission* and *directional reception* – that is, the signal is sent out in all directions (to anyone within earshot) and the listener can identify the direction the sound is coming from and thus identify the sender; in contrast, programming languages feature directional transmission in the form of computational input, but produce broadcast reception. The feature of *interchangeability* exists in human languages – a typically developing person could both receive and broadcast the same message, whereas computers and their programmers cannot switch roles. However, as mentioned, programming languages also convey information between human beings. Tremblay and Sorenson (1985, p.74) consider this to be “the most important goal of a programming language”; a survey of over 780 programmers found that 95% agreed that understanding existing code is a significant part of their job (Cherubini, Venolia, DeLine, & Ko, 2007). Like natural languages, computer programming languages allow people to collaborate and share ideas.

Although there are differences between programming languages and natural languages, there are also many similarities. Weinberg (1971) concedes that computer programming languages appear to “live up to or exceed natural language[s]’ in the remaining nine areas identified by Hockett (1960). For example, like natural languages, programming languages are arbitrary – there is no direct connection between the word and its meaning (an aspect that proves confusing for novices, as discussed below); both programming and natural languages also have duality of patterning (they can be broken down into small parts which can be recombined to form new meaning).

Norman (2008) argues that programming languages technically meet “the three most important criteria” for languages. First, a language must be meaningful; whilst natural languages use words and sentences to convey meaning, programming languages achieve this via objects, functions and relations. Second, a language should allow displacement, that is, communication about things that are not immediately present; natural language achieves this with concepts such as yesterday or somewhere else, and programming languages systematically convey instructions for the future and reference data from the past. Third, a language must be productive, allowing the expression of ideas that have never been expressed before; computer programming languages are particularly strong in this respect.

Few studies have systematically compared natural and programming languages. Kokol, Podgorelec, Zorman, Kokol, and Njivar (1999) provided a rare exception. The authors compared the complexity of natural language texts to computer programs using long-range power law correlations (LRCs). The authors found that the mean $\alpha$ values – which are thought to measure the complexity or information content of the text – were significantly higher for programming languages. This difference was attributed to three key differences: first, the formality of programming languages, which forces order and reduces randomness; second, the ambiguity of natural languages, which creates randomness and disorder; third, the significantly larger vocabularies that feature in natural language contribute more to randomness than complexity. In short, computer programming languages were found to be more efficient in communicating complexity. However, the presence of LRC in both natural and programming languages indicates that they are derived from common (as yet unidentified) laws.

We have established that there are shared features between natural languages and programming languages; however, it would be ridiculous to suggest that the acquisition of programming languages is similar to an infant's acquisition of language. Instead, we will consider similarities between the later acquisition of second languages and the difficult task of acquiring the fluent use of computer programming languages. One striking similarity is the problem of *language transfer*. This term is traditionally used to describe the influence of the native language (L1) on the acquisition of a subsequent language (L2); Pavlenko and Jarvis (2002) have argued that this effect is in fact bidirectional. Cross-linguistic interference is the negative side of language transfer, and can occur at the lexical, semantic and also at conceptual level (Pavlenko, 2009). At the lexical level, errors may occur when speakers erroneously assume that similar-sounding words share meaning *(false cognates)*, when words from the wrong language are unconsciously inserted into sentences *(unintentional language switches)*, or when speakers inadvertently blend words from two languages to create new words – for example, a Swedish-English bilingual may create the word *clothers* from the English *clothes* and the Swedish *klä der* (Jarvis, 2009). At the conceptual level, Boroditsky (2001) found that Mandarin-English bilinguals showed a bias towards thinking about time in terms of vertical metaphors (which is how time is conceptualised in the Chinese language) even when completing a task in English (which conceptualises time in terms of horizontal metaphors); the strength of this bias was positively correlated with age of English acquisition and the extent to which participants reported thinking in English.

Studies of the errors made by novice programmers indicate that knowledge of natural language influences the way programming languages are understood by novice and intermediate programmers (Witschital, 1995). Soloway and Spohrer (1989) videotaped interviews with novice programmers solving problems and documented examples of bugs caused by natural language transfer. For example, in the English language lexicon, instructions such as *while* imply continuous checking of the condition ("while x is happening, do y"); the novices therefore erroneously expected similar results from the while loop in Pascal (in which the condition is only checked at discrete points of time). This supports the findings of an earlier study (Soloway *et al*., 1981) in which 34% of the students in an introductory programming courses reported this "*while demon"* misconception. Another example from Soloway and Spohrer (1989) concerned *if-then-else* commands. In natural language, this implies a looping construct – for example, "*If you find a door unlocked then lock it"*; the novice therefore confuses lexical similarity between the two languages with functional similarity. Negative transfer has also been identified in the naming of variables; Sleeman, Putnam, Baxter, and Kuspa (1986) found that students frequently erroneously believed that assigning meaningful names (in natural language) to variables in computer programming would have somehow aid the computer's understanding of the program. These errors are very similar to the false cognates errors experienced by bilinguals and second language learners.

Language transfer can have positive as well as negative effects, as knowledge about the way language works can aid the acquisition of additional languages. Metalinguistic awareness is knowledge of the structural components of language, for example, understanding of the arbitrary connection between symbols and their referents (Yelland, Pollard, & Mercuri, 1993). Childhood bilingualism has been linked to the advanced development of metalinguistic awareness. For example, Ianco-Worrall (1972) asked Afrikaans/English bilingual children whether it was possible to swap the names for cow and dog; twice as many bilingual children understood that this was possible, compared to their monolingual peers. Similarly, Ben-Zeev (1977) found that Hebrew-English bilinguals performed better at symbol substitution tasks compared monolinguals, suggesting that experience of more than one language system left children "freer to abandon the rules of a particular language system for a different set of rules where necessary".

The relationship between computer programming and metalinguistic ability has attracted less empirical attention. However, computer programming languages not only provide a new lexicon, as natural language does, but also necessitate the programmer's active involvement in the construction of that lexicon via the naming of variables (Détienne, 2002); therefore it is reasonable to predict that there is a relationship between programming and metalinguistic ability. Some support for this assumption can be found in studies of the effects of programming instruction on children with language or literacy deficits. Lehrer and DeBernard (1987) found that language-impaired children who received Logo programming instruction performed better in tests of perceptual-language skills, compared to children who used commercially available learning software or teacher-led instruction. Peppler and Warschauer (2011) observed how "Brandy", a nine-year-old girl with learning disabilities, learnt to program using Scratch – an accessible programming language for children, in which programs are built from

Lego-inspired graphical programming blocks (Resnick *et al*., 2009) – before she could read or write. The authors, who followed Brandy over a two-and-a-half year period, linked Brandy's developing programming literacies to marked improvement in her traditional print literacy: "Working with Scratch seemed to illuminate for Brandy the mechanics of language and stimulate her metalinguistic awareness of how language operates, as she made connections between the Scratch programming language and her spoken (and increasingly written) English language" (p.32).

In summary, descriptions of the learning experiences of novice programmers indicate that they experience language transfer effects. The errors they make suggest that in order to successfully master computer programming languages, novices must develop the ability to successfully inhibit transfer from their spoken languages. In the bilingual literature, it is generally agreed that the mechanism responsible for the successful management of two or more languages – and thus the avoidance of negative transfer – is some domain-general aspect of executive control. It is therefore logical to hypothesise that this same mechanism is employed by computer programmers to facilitate the successful management of two very different language systems. The exact specification of this mechanism is currently unclear – a promising theory suggests a conflict monitoring system, in which the dorsal ACC plays a key role – but its existence is indicated by the presence of a bilingual advantage in executive control tasks. The present study therefore investigates the performance of computer programmers on these tasks in order to ascertain whether computer programmers benefit from the "bilingual advantage".

## Section III: The present study

Literature searches indicate that to date, there has been little systematic investigation of executive control abilities in computer programmers. Studies that have considered the impact of computer programming on specific aspects of executive function have focused on the role of working memory (e.g. Bergersen & Gustafsson, 2011). No studies comparing the cognitive effects of computer programming to the cognitive effects of bilingualism have been found. The present study aimed to contribute towards filling this gap in the literature.

The performance of computer programmers on two executive control tasks that have been found to demonstrate the bilingual advantage was compared to that of monolingual, age-matched controls. The aim was to demonstrate whether the computer programmers showed a similar advantage. The Attention Networks Task (Fan, McCandliss, Sommer, Raz, & Posner, 2002) is a computer-based task designed to evaluate three attention networks: alerting (achieving and maintaining an alert state), orienting (selecting information from sensory input), and executive attention (monitoring and resolving conflict). Costa *et al*. (2008) compared the ANT performance of Spanish-Catalan bilinguals (mean age 22 years) to monolingual controls; the former recorded faster global reaction times (RTs) and exhibited less interference in conflict conditions. Importantly, using a lateralised version of the ANT (which was designed to provide additional information about the interhemispheric organisation of the attention networks), Tao *et al*. (2011) found that early but less-balanced bilinguals had significantly higher global RTs, but that late proficiency but more balanced bilinguals showed an advantage in conflict resolution specifically. A computer-based version of the Stroop colour-word test was also included to provide a second measure of executive control; Bialystok *et al*. (2008) found that bilinguals displayed smaller Stroop effects than monolinguals (though there was no difference in reaction times).

The study involved two groups of programmers, all of whom (like Tao *et al.*'s late proficiency bilinguals) acquired programming languages during adolescence. In order to provide an indication of the length of programming experience required to obtain any observed executive control advantages, two different age groups were used: young adults, employed professionally as programmers, with at least five years' experience of programming; and adolescents who programmed regularly but had less programming experience. It was initially hoped to include a third group, children aged 8-11 who had 6 months' programming experience, in order to provide further insight into any cut-off point; however, last minute recruitment difficulties precluded this.

The hypotheses for the study were as follows: 1. Programmers would record faster global RTs than controls on both executive control tasks. 2. Programmers would be less affected by conflict conditions than controls in the executive control tasks. 3. The difference between programmers and controls would be greater in the older age group, due to greater experience of computer programming.

# Method

## Participants

A total of 40 monolingual, English-speaking adolescents and young adults participated in the study. Participants were grouped by age (adolescent or young adult) and programming experience (programmer or non-programmer). The mean age of the young adults' groups (programmers: $N = 10$, $M = 24.13$, $SD = 0.82$; non-programmers: $N = 10$, $M = 23.74$, $SD = .41$) were not statistically different from each other; nor were the adolescents' groups (programmers: *M* = 16.05, *SD* = 1.17; non-programmers: *M* = 16.39, *SD* = 1.62).

The young adult programmers were recruited via contacts at digital agencies in London. All were currently employed as professional computer programmers; they had had an average of *M* = 7.8 years' programming experience, *SD* = 2.23; 4 reported coding "every day", 6 coded "most days". The group reported regularly using an average of *M* = 3.34 programming languages; the most commonly used languages were JavaScript (10), Ruby (8) and PHP (6). The group comprised 10 males and 1 female. The young adult non-programmers were recruited via word of mouth. There were 10 males and one female; all were currently employed in office-based professions involving daily computer use, but they had no experience of computer programming.

In the adolescent programmers' group, 6 participants were recruited via an advert in the newsletter of Young Rewired State, an organization that runs annual events for young programmers. It had been hoped that all young programmers would be recruited in this way; however the timing (summer term), the age group (GCSEs, AS-levels) and the monolingual requirement (4 additional adolescents expressed interest but were ineligible due to being bilingual) hindered recruitment. Therefore an additional 4 adolescents were recruited via an IT teacher at a school in Kent, where computing GCSE is taught. The group comprised 6 males and 4 females with an average of *M* = 2.7 years' programming experience, *SD* = 1.70; 4 reported coding "most days", 6 coded "weekly". The group reported regularly using an average of *M* = 2.7 programming languages; the most commonly used languages were Javascript (6), Java (4), C# (4) and Scratch (4). The control group comprised 10 adolescents, 6 males, 4 females, recruited via family and friends.

**Ethics.** All participants received information sheets, tailored to their age and programming group, containing details of the study. This explained that people who speak two or more languages perform better at some tasks that involve quickly deciding what to pay attention to and what to ignore; the aim of the study was to find out whether people who regularly use computer programming languages are also better at these tasks. All participants gave written consent to participation prior to taking part. In addition, written parent or guardian consent was obtained for all under 18s. The research received ethical approval under the Institute of Education ethics procedures. Participants were not paid for taking part.

## Design

The research employed a quasi-experimental factorial design, with between-subject factors of programming experience and age. Participants were first asked to complete the Wechsler Abbreviated Scale of Intelligence (Wechsler, 1999; 2-subtest version) in order to establish that the age-matched groups did not differ in IQ. In order to test the experimental hypothesis, executive control was then tested by means of the Attentional Networks Task (ANT, Fan *et. al*, 2002) and the Stroop colour-word task (Stroop, 1935).

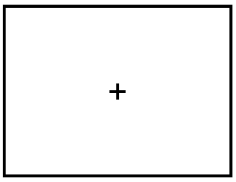
### Materials.

*Wechsler Abbreviated Scale of Intelligence (WASI).* In order to compare IQ across the groups, participants completed the 2-subtest version of the Wechsler Abbreviated Scale of Intelligence (Wechsler, 1999). First, participants completed the Vocabulary subtest, which assesses the vocabulary and lexical knowledge of the participant. Second, participants completed the Matrix reasoning subtest, which assesses the fluid intelligence of the participant. Raw scores were converted into t-scores according to the participant's age, and combined in order convert total t-scores into FSIQ-2 score.

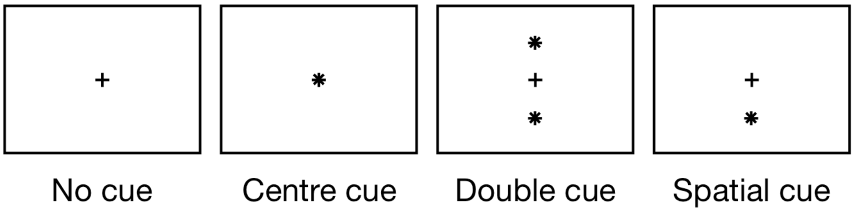
*Attentional Networks Test (ANT)***.** The first of two tasks selected to assess executive control was the Attention Network Test, which was adapted from Fan *et al.* (2002). As mentioned earlier, the ANT is a computer-based task designed to evaluate three attention networks: alerting (achieving and maintaining an alert state), orienting (selecting information from sensory input), and executive attention (monitoring and resolving conflict).

After reading on-screen instructions, participants were presented with a series of arrows on a screen, pointing either left or right. Their task was to identify the direction in which the target arrow was pointing by pressing the corresponding arrow key on the keyboard, as quickly and accurately as possible. In order to assess executive control, target arrows were presented either alone ("neutral flanker type") or flanked by arrows pointing in the same direction ("congruent flanker type") or in the opposite direction ("incongruent flanker type") (see Box 2, Step 3). Arrows were presented either above or below a fixation point in the centre of the screen. Alerting and orientating networks were assessed through the use of cues prior to the presentation of the stimulus: in order to manipulate alertness, participants received either a cue comprising two stars ("double cue") or no cue; in order to assess orienting effects, participants received either a cue indicating where on the screen the arrows would appear ("spatial cue") or a cue in the centre of the screen which gave no clues about where the arrows would appear ("central cue") (see Box 2, Step 2).

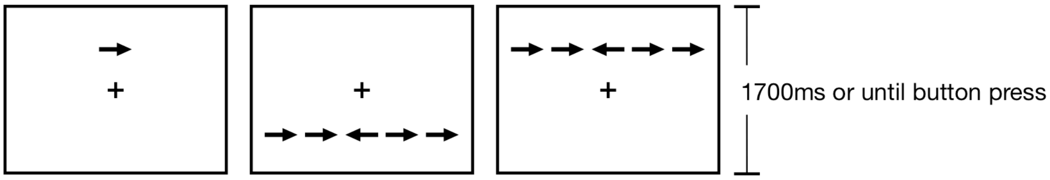
**Box 2: The adaption of the Attention Networks Task used in the present study**



**Step 1: fixation point.** At the beginning of each trial, participants viewed a fixation point for 400 ms.



**Step 2: cue.** In no cue trials, no cue was presented. In centre cue and double cue trials, either one or two cues (*) were presented; these did not indicate the location in which the arrows would appear. In spatial cue trials, the location in which the cue appeared indicated where the target arrow would appear.



**Step 3: arrows.** Target stimulus was then presented either above or below the fixation point for 1700ms or until a response was given. In neutral trials, the target arrow appeared alone. In congruent trials, the target arrow was flanked by arrows facing in the same direction; in incongruent trials, flanker arrows faced in the opposite direction.

In total, there were 12 different types of trial: flanker type (congruent, incongruent or neutral) x cue type (no cue, centre cue, double cue, spatial cue). Each type of trial was repeated 4 times, creating a total of 48 trials. Trials were presented in a random order. Each trial ran as follows: 1. A fixation point (+) appeared in the centre of the screen for 400ms; 2. A cue (or no cue) was presented for 150ms; 3. The target stimuli was presented (with or without flankers), disappearing either when the participant responded or after 1700ms. (See Box 2, Step 3).

The ANT was programmed using Paradigm experimental software; the software recorded response time and accuracy for each trial. All stimuli were presented in black font, on a light blue background. The fixation point and cues were presented in Microsoft Sans Serif font at 27.75pt. Target and flanker arrows were created using the font Wingdings 3 at 36pt; these were presented either above, or below the fixation point. Prior to the actual task, a practice run of 12 trials was conducted in order to ensure that participants understood the task. There were no difficulties in task comprehension

*Stroop Colour-Word Task.* The second task employed to assess executive control was a computer-based version of the Stroop colour-word task (Stroop, 1935). The task assesses participants' ability to selectively attend to stimuli, filtering out misleading information. On-screen instructions explained to participants that they would see a series of words written in different colours. Their task was to identify the colour in which the word was written by pressing one of two keys on the keyboard: **RED** and **BLUE** mapped onto the left arrow key; **GREEN** and **YELLOW** mapped on the right arrow key.

Each word belonged to one of three trial types: congruent, incongruent or neutral. In congruent and incongruent trials, the words were RED, YELLOW, GREEN, BLUE. In congruent trials, the colour of the words matched their semantic meaning: **RED**, **YELLOW**, **GREEN**, **BLUE**. In incongruent trials, the meaning of the colour word conflicted with the target colour, resulting in semantic conflict e.g. **RED**, **YELLOW**, **GREEN**, **BLUE**. All semantically incongruent condition also required an incongruent response; no incongruent colour-words were presented that mapped onto the same response key as the target colour. The neutral condition comprised four non-colour words: DOG, JUMP, KNIFE and FLOWER. These words are not semantically associated with any of the target colours, but their character lengths correspond to the character lengths of the four colour words. Each neutral word was presented three times, in a different colour each time (red, blue, yellow or green, selected at random).

The Stroop colour-word task was programmed using Paradigm experimental software; the software recorded response time and accuracy for each trial. Stimuli comprised a series of words presented in one of four target colours – red, blue, green and yellow – using Courier New font in bold at 36 point. These words were presented in the centre of the screen, on a Gainsboro (light grey) background.

A total of 12 trials were conducted in each condition, making a total of 36 trials. Trials were presented in a random order. Each word disappeared after 2000ms or as soon as a response was given. Prior to the actual task, a practice run of 8 trials was conducted in order to ensure that participants understood the task. There were no difficulties in task comprehension.

## Procedure

Due to the disparate geographic location of participants in the experimental groups (Greater London, Birmingham, St Albans, Milton Keynes, Manchester, Cheltenham, Kent, West Sussex), the researcher visited participants at their homes (adolescents), work places (young adults) or alternative suitable locations as required, and the testing took place in a quiet, well-lit room. For practical reasons, there were differences in the times and days that the different age groups were seen: adolescents were seen at weekends or holidays, whereas young adults were seen towards the end of a working day. The order of testing was: WASI Vocabulary, WASI Matrix Reasoning, ANT, Stroop colour-word task. The executive function trials were all presented on a

Toshiba Equium L40 156 laptop computer with a 15 inch screen, with a resolution of 1280 x 800 pixels. At the end of the experiment, participants were debriefed and offered the opportunity to ask questions.

# Results

Results from the experiment were analysed as follows. ***WASI:*** In order to identify any significant IQ differences that might influence results, a two-way ANOVA was first conducted on FSIQ-scores, with age and programming experience as the independent variables. ***ANT:*** Group performances on the ANT were then compared. Response times analysed via a three-way mixed ANOVA with a within-participant factor of congruency (3 levels: congruent, incongruent, neutral) and two between-participant factors: age group (2 levels: adolescent or young adult) and programming experience (programmer or non-programmer). Conflict effect was calculated (incongruent trials RT – congruent trials RT) and a two-way ANOVA performed with age and programming experience as the independent variables. Kruskal-Wallace tests were used to compare overall % errors and conflict effect ERR (errors on incongruent trials – errors on congruent trials). ***Stroop colour-word task:*** Group differenced in performance on the Stroop colour-word task were then analysed. A Kruskal-Wallis test was used to compare response times of the four groups in each of the three trial types (congruent, incongruent and neutral). The Stroop effect was calculated (incongruent RT – congruent RT); a Kruskal-Wallis test was used to compare group differences in the size of the Stroop effect. A MANOVA was used to compare differences in facilitation effect (congruent RT – neutral RT) and cost (incongruent RT – neutral RT). Finally, Kruskal-Wallis tests were used to compare group differences in global errors and in Stroop error effects (incongruent ERR% - congruent ERR%).

**WASI**

Mean FSIQ-2 scores for each of the four groups are presented in *Table 1*. Exploratory data analysis indicated that the data met the assumptions required for parametric testing; there was one extreme (low) outlier in the adolescent programmers group but this had a minimal effect on the mean for the group so it was not removed from the sample.

**Table 1.**

*Mean FSIQ-2 Scores And Standard Deviations For Adolescent and Young Adult Programmers And Non-Programmers.*

| Age group | Programming experience | Mean FSIQ-2 | SD |
|---|---|---|---|
| Adolescents | Programmer | 112.30 | 9.44 |
| | Control | 113.40 | 6.93 |
| Young adults | Programmer | 119.50 | 7.68 |
| | Control | 118.30 | 5.23 |

A two-way ANOVA was conducted with FSIQ-2 score as the dependent variable; the factors were age and programming experience. This revealed a significant difference in age, $F(1, 36) = 6.28, p < .05$, partial $\eta^2 = .15$; the young adults had higher FSIQ-2s scores than the adolescents. There was no significant difference in programming experience, $F < 1$, and no significant interaction between age and programming experience, $F(1, 36) = .23, p > .05$.

## Attentional Networks Test

### ***Executive control network.***

*RT analysis.* Response time (RT) analysis focused primarily on participants' overall scores in each flanker condition – congruent, incongruent and neutral – as this monitors the executive control network, in which bilingual advantage has been found.

Response times (RTs) were analysed for each participant's responses to each of the 48 trials. RTs for error trials and no response trials were excluded (3.70% of all responses). Median RTs were then calculated across each flanker condition (congruent, incongruent, neutral).

First, exploratory data analysis was performed. The Kolmogorov-Smirnov test indicated that the data for the adolescent non-programmers group in the congruent condition, $D(10) = .276, p <. 05$, were not normally distributed; however, the more sensitive Shapiro-Wilk test was not significant: $D(10) = .883, p > .05$. Levene's Test of Homogeneity of Variance was significant in the incongruent condition, $F(3, 36) = 3.39$, $p < .05$. Examination of the boxplots revealed extreme scores for case 25 in both congruent and incongruent conditions; these scores had a substantial effect on the mean RT of the adolescent non-programmers' group and were therefore removed from the analysis. Other outliers (cases 14, 19 and 29 in the congruent condition; 14, 19 in the incongruent condition; 3, 5 in the neutral condition) did not substantially affect the mean and were therefore retained. *Table 2* presents median RTs and SDs by group and flanker condition (congruent, incongruent, neutral) following the removal of case 25.

**Table 2**.

*Mean Response Times and Standard Deviations for Adolescent and Young Adult Programmers and Controls in the Attention Networks Task*

| | Adolescents | | | | Young Adults | | | |
|---|---|---|---|---|---|---|---|---|
| | Programmers | | Controls | | Programmers | | Controls | |
| Trial type | RT (ms) | SD | RT (ms) | SD | RT (ms) | SD | RT (ms) | SD |
| Congruent | 483.9 | 65.0 | 520.1 | 50.8 | 474.1 | 43.6 | 551.4 | 87.1 |
| Incongruent | 563.5 | 49.5 | 611.4 | 109.8 | 524.4 | 33.7 | 613.1 | 51.4 |
| Neutral | 456.0 | 466.5 | 503.2 | 62.7 | 454.3 | 32.1 | 528.8 | 57.1 |

A three-way mixed ANOVA was conducted with a within-participant factor of congruency (3 levels: congruent, incongruent, neutral) and two between-participant factors: age group (2 levels: adolescent or young adult) and programming experience (programmer or non-programmer). This showed a main effect of congruency, $F$ (2, 70) = 93.56, $p < .001$, partial $\eta^2$ = .73. This was due to all groups responding more slowly in the incongruent condition than in the neutral and congruent conditions, as illustrated in *Fig. 1*. The main effect of programming experience was also significant, $F$ (1, 35) = 8.01, $p < .01$, partial $\eta^2$ = .19. This was as a result of programmers responding more quickly than non-programmers in all conditions. The effect of age was not significant, $F < 1$; as *Fig. 1* illustrates, young adult programmers had the global RTs, and young adult non-programmers had the slowest global RTs. None of the interactions were significant (congruency and age; congruency and programming; and congruency, age and programming), $F < 1$.

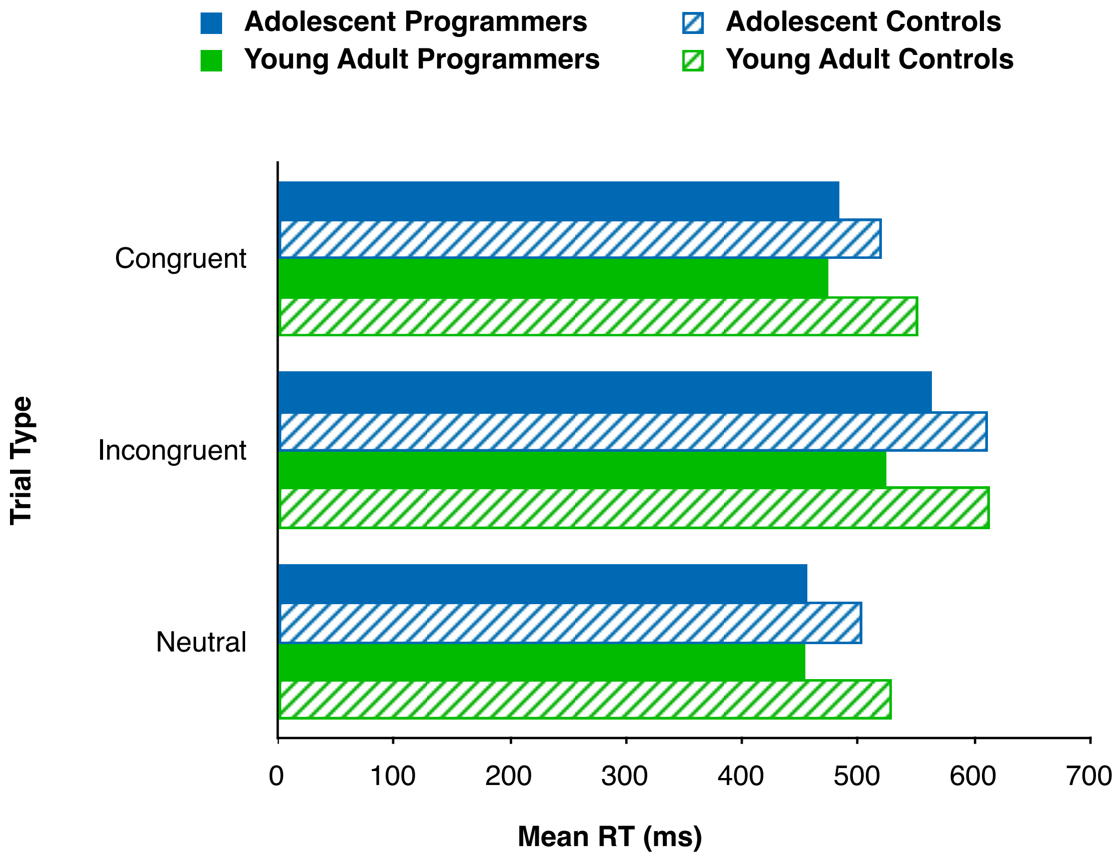


***Figure 1.*** **Line graph showing how age and programming experience affected response times in congruent, incongruent and neutral trials in the Attention Networks Task. Both groups of programmers were faster than non-programmers.**

Conflict effect RT was calculated by subtracting RT in the congruent condition from RT in the incongruent condition. A two-way ANOVA was performed with conflict effect RT as the dependent variable; age and programming experience were the independent variables. The results were not significant, $F < 1$; there was no significant difference in the degree to which the groups were disadvantaged in the incongruent conditions.

*Error analysis.* Table 3 shows the mean % error rates for all four groups in congruent, incongruent and neutral conditions, as well as the total error rates. Three participants were excluded from the analysis due to extreme high scores (cases 7, 19, and 25) that had a substantial effect on group means.

**Table 3.**

*ANT Task: Mean Percentage Error Rates By Group and Flanker Type.*

| | Adolescents | | | | Young Adults | | | |
|---|---|---|---|---|---|---|---|---|
| | Programmers | | Controls | | Programmers | | Controls | |
| Trial type | ERR% | SD | ERR% | SD | ERR% | SD | ERR% | SD |
| Congruent | 0 | 0 | 0 | 0 | 0.7 | 2.1 | 0 | 0 |
| Incongruent | 8.3 | 8.3 | 7.6 | 7.1 | 9.7 | 9.2 | 5.0 | 4.9 |
| Neutral | 0 | 0 | 0.6 | 2.0 | 1.3 | 4.0 | 3.1 | 1.4 |
| Overall | 2.8 | 2.8 | 2.8 | 2.6 | 3.9 | 3.2 | 2.7 | 2.0 |

Exploratory data analysis indicated that the distribution of the total error rates deviated significantly from normality in the programming, $F(18) = .28, p < .01$, and non-programming, $F(19) = .25, p < .05$, groups. Therefore, a Kruskal-Wallace test was used to compare all four groups: adolescent programmers, adolescent non-programmers, young adult programmers and young adult non-programmers; the difference between the groups was not significant, $H(3) = 1.06, p > .05$.

Conflict effect ERR was calculated by subtracting ERR% in the congruent condition from ERR% in the incongruent condition. The Kolmogorov-Smirnov test indicated that although the data were normally distributed in the professional programmers' group, $D(10) = .18, p > .05$, they were not normally distributed in the remaining three groups: adolescent programmers, $D(10) = .36, p < .05$, adolescent non-programmers $D(10) = .35, p < .05$ and professional non-programmers, $D(10) = .25, p < .05$. A non-parametric Kruskal-Wallis test was used to compare the means; the results were not significant, $H(3) = 2.01$, p = .56.

## Stroop Colour-Word Task

Response times (RTs) were analysed for 12 trials of each condition (congruent, incongruent and neutral). RTs for error trials and no response trials were excluded (3.06% of all responses). Median RTs were then calculated for each participant in each condition. Initial examination of the revealed the presence of a small Stroop effect for the adolescent controls and young adult programmers, and a small negative Stroop effect for adolescent programmers and young adult controls; a surprising result. It was hypothesized that, due to the random presentation of the stimulus, there was an uneven distribution of trials in which the target colour was the same as the proceeding trial. Further examination of the data revealed that this was indeed the case. Participants varied in the number of these types of trial they experienced, and in whether these trials occurred on congruent, incongruent or neutral trials; this negated the Stroop effect. Therefore, RTs for these trials were excluded (23.06% of all responses) and median RTs recalculated for each participant. Mean RTs and SDs are presented in *Table 4*.

*RT analysis.* As Table 4 illustrates, young adult programmers displayed the slowest RTs across all three conditions. Exploratory data analysis indicated that the data violated the assumption of homogeneity in congruent, $F(3, 35) = 10.09, p < .001$, incongruent, $F(3, 35) = 3.52, p < .05$, and neutral trials, $F(3, 35) = 5.55, p < .05$. A Kruskal-Wallis test was used to compare the four groups' RTs in each of the three conditions. This revealed a significant difference in reaction times between groups in the congruent condition, $H(39) = 12.45$, $p < .05$; the young adult programmers ($M$ = 781.3, $SD$ = 344.7) were much slower than the young adult controls ($M$ = 591.1, $SD$ = 114.3), while the adolescent programmers ($M$ = 546.4, $SD$ = 64.2) were slower than the adolescent controls ($M$ = 471.7, $SD$ = 55.5), who were the fastest group overall. The professional programmers were also the slowest group in the incongruent and neutral conditions, but this difference was not significant, $p > .05$.

**Table 4.**

*Response Times For Adolescent and Young Adult Programmers and Non-Programmers in the Stroop Colour-Word Task.*

| | Adolescents | | | | Young Adults | | | |
|---|---|---|---|---|---|---|---|---|
| | Programmers | | Controls | | Programmers | | Controls | |
| Trial type | RT (ms) | SD | RT (ms) | SD | RT (ms) | SD | RT (ms) | SD |
| Congruent | 546.4 | 64.2 | 471.7 | 55.5 | 781.3 | 344.7 | 591.1 | 114.3 |
| Incongruent | 569.9 | 108.9 | 545.1 | 80.7 | 739.0 | 265.0 | 642.9 | 179.2 |
| Neutral | 577.9 | 108.9 | 529.5 | 58.5 | 715.2 | 243.8 | 656.6 | 177.9 |

The Stroop effect is calculated by subtracting the participant's median score in the congruent trials from their median score in the incongruent trials. Due to the presence of interference in the incongruent trials, we would expect participants to record slower response times, thus we would predict a positive Stroop effect score for all groups. As Fig X shows, adolescent programmers ($M$ = 23.42, $SD$ = 82.14), adolescent non-programmers ($M$ = 73.45, $SD$ = 51.48) and professional non-programmers ($M$ =

51.82, *SD* = 89.51) all demonstrated a positive Stroop effect; this reflects higher RTs in incongruent trials than in congruent trials. However, professional programmers demonstrated a negative Stroop effect (*M* = -11.57, *SD* = 76.27); contrary to predictions, they responded more quickly in incongruent than congruent trials.

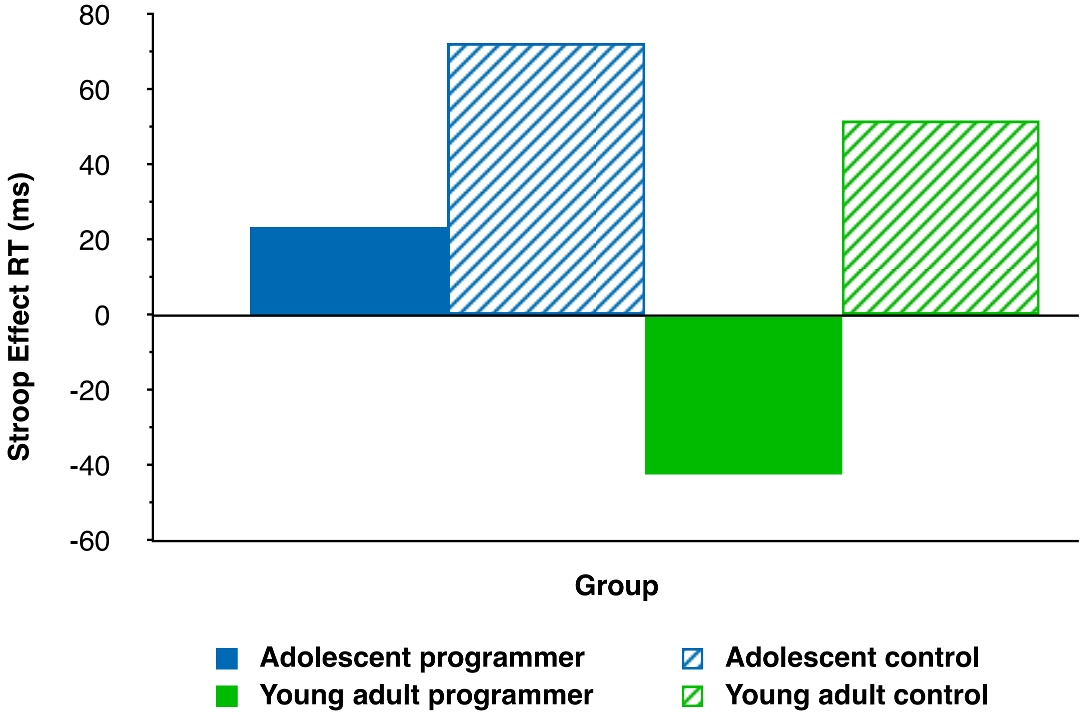


***Figure 2.*** **Bar chart showing the Stroop effect by group. Stroop effect was calculated by subtracting mean RT in congruent trials from mean RT in incongruent trials.**

Exploratory data analysis on Stroop effect RT by group indicated that data in the adolescent programmers' group, *D*(10) = .266, *p* <. 05, and the professional programmers' group, *D*(10) = .317, *p* < .05, were not normally distributed. Therefore, the groups were compared using a Kruskal-Wallis test; the difference between the groups was not significant, *H*(39) = 7.244, *p* > .05.

Another way of analysing the data is in terms of facilitation (the use of the additional helpful information in the congruent trials) and cost (the extra effort required to suppress the misleading information in the incongruent trials) relative to scores in the neutral trials. Mean RTs for Stroop facilitation were calculated by subtracting scores in the congruent trials from scores in the neutral trials. Since we would expect the congruent RTs to be faster than neutral RTs, due to the presence of helpful information, we would expect Stroop facilitation scores to be positive. Mean RTs for Stroop cost were calculated by subtracting incongruent trial RTs from neutral trial RTs. Since we would expect the incongruent RTs to be slower than neutral RTs, we would expect this score to be negative. 1 extreme outlier in the non-programmers' group (case 38), which was significantly affecting the mean and standard deviation in that group (with case 38, *M* = 13.70, *SD* = 161.20, without case 38, *M* = -29.16, *SD* = 86.58), was removed from the analysis. The resulting RTs and standard errors for facilitation and cost by group are presented in Fig 3. As we can see, there were two surprising trends in the data. First, the adolescent programmers group recorded a positive Stroop cost score (*M* = 8.0, SD = 80.03), which means that they responded more quickly in incongruent trials than in neutral trials; the adolescent controls (*M* -15.65, *SD* = 69.65), young adult programmers (*M* = -23.87, *SD* = 85.81) and young adult controls (*M* = -29.20, *SD* = 84.57) all recorded negative scores, as would be expected. Second, and even more surprisingly, the young adult programmers recorded a negative Stroop facilitation score (*M* = -66.10, SD = 131.57), which means that they were negatively affected by congruent trials in comparison to neutral trials. The young adult controls (*M* = 35.08, *SD* = 83.33), adolescent programmers (*M* = 31.48, *SD* = 110.60), and adolescent controls (*M* = 57.80, *SD* = 70.74) all recorded positive facilitation scores, as expected.

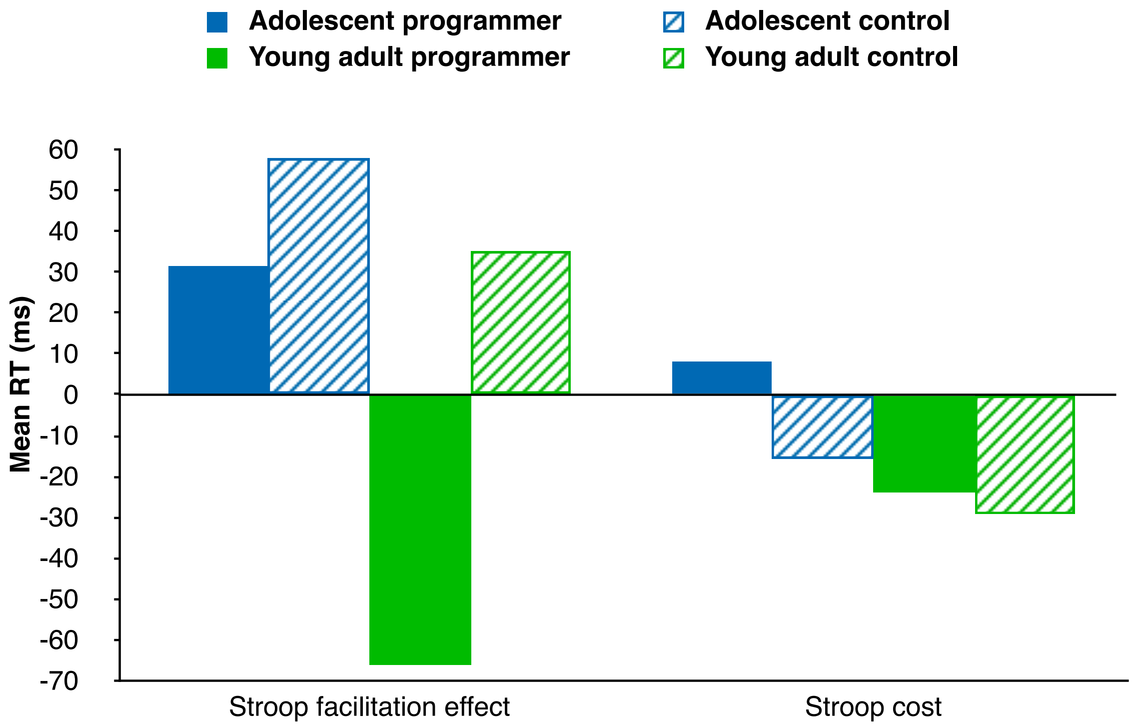


***Fig 3.*** **Bar chart showing Stroop facilitation effect (Congruent RT – Neutral RT) and Stroop cost (Neutral RT – Incongruent RT) by group.**

Kolmogorov-Smirnov and Shapiro-Wilk tests confirmed that data was normally distributed across all groups, Levene's Test of Homogeneity of Variance was not significant, and there were no outliers; the data met the criteria for parametric testing. A MANOVA was conducted with Stroop facilitation effect RT and cost effect RT as the dependent variables and *group* as the independent variable (4 levels: adolescent programmer, adolescent control, young adult programmer, young adult control). Using the Roy's largest root statistic, this effect was significant, $F(3, 34) = 3.71$, $p < .05$, $\Theta = .33$; however Pillai's trace, Wilks' Lambda and Hotelling's trace returned non-significant results, $p > .05$. Univariate analyses indicated that the effect of group on Stroop facilitation effect was approaching significance, $F\ (3, 34) = 2.84$, $p = .053$, partial $\eta^2 = .200$, reflecting atypical direction of the effect in young adult programmers; the effect of group on Stroop cost was not significant, $F < 1$.

***Error analysis*** Percentage error rates in congruent, incongruent and neutral trials, overall percentage errors, and Stroop error effect (ERR% congruent trials – ERR% incongruent trials) were calculated and are presented in Table 5. Participants 8 and 12 were excluded from the analysis due to extremely high error rates. As Table 5 shows, young adult controls (*M* = 2.0, *SD* = 2.0) followed by adolescent programmers (*M* = 2.5, *SD* = 1.7) made the fewest errors; adolescent controls (*M* = 5.8, *SD* = 4.0) made the most errors, followed by young adult programmers (*M* = 5.2, *SD* = 4.0).

**Table 5.**

*Error Rates and Stroop Error Effect for Adolescent and Young Adult Programmers and Non-Programmers in the Stroop Colour-Word Task*

| | Adolescents | | | | Young Adults | | | |
|---|---|---|---|---|---|---|---|---|
| Trial type | ERR% | SD | ERR% | SD | ERR% | SD | ERR% | SD |
| Congruent | 1.8 | 3.6 | 5.8 | 6.9 | 7.4 | 8.8 | 1.9 | 3.7 |
| Incongruent | 2.8 | 4.2 | 6.7 | 5.3 | 4.6 | 8.4 | 3.7 | 6.1 |
| Neutral | 2.8 | 4.2 | 5.8 | 6.9 | 3.7 | 6.1 | 0 | 0 |
| All | 2.5 | 1.7 | 5.8 | 4.0 | 5.2 | 4.0 | 2.0 | 2.0 |
| Stroop effect | 4.6 | 8.5 | 0.8 | 8.3 | -2. 8 | 12.5 | 1.9 | 8.1 |

*Global errors.* Exploratory data analysis of total error rates indicated that the data were not normally distributed in three of the four groups: adolescent programmers, $D(8) = .327, p <. 05$, adolescent controls, $D(10) = ..276, p <. 05$, and young adult controls, $D(9) = .272, p <. 05$. A Kruskal-Wallis test indicated that there was a significant difference between the groups, $H(36) = 8.51, p < .05$; however post-hoc Mann Whitney tests with Bonferroni corrections were not significant, $p >.008$.

*Stroop effect errors.* Exploratory data analysis of the Stroop error effect indicated that the data were not normally distributed in the adolescent programmers' group, D(9) = .375, p = .001. A Kruskal-Wallis test indicated that there was a significant difference between the groups, $H(36) = 8.51, p < .05$. This reflected the unusual negative Stroop error effect recorded by the young adult programmers (M = -2.78, SD = 12.5), indicating that they made more errors in congruent trials than incongruent trials; in contrast, the adolescent programmers (M = 4.63 SD = 8.45), adolescent controls (M = 0.84, SD = 8.29) and young adult controls (M = 1.85, SD = 8.10) made more errors in incongruent trials.

# Discussion

The goal of the present study was to investigate whether the "bilingual advantage" was also found in computer programmers. The study was motivated by a hypothesis that the that same mechanism used by bilinguals to manage two separate languages may also be utilised by computer programmers, in order to prevent negative transfer between computer programming languages and natural languages. The study used two tests of executive control in which bilinguals have been found to outperform their monolingual peers: the Attention Networks Task (ANT) and the Stroop colour-word task. It was predicted that computer programmers would respond to these tasks more quickly than their monolingual peers. More tentatively, it was predicted that a reduced conflict effect might also be found in the computer programming groups.

All of the computer programmers who participated in the study acquired their computer programming languages during adolescence; importantly, bilinguals who acquired their second language during adolescence have been found to exhibit the bilingual advantage. Two groups of programmers, with differing levels of experience, were studied in order to provide an indication of the extent of programming language experience required to produce an advantage. In order to provide a baseline comparison, two groups of age-matched and IQ-matched controls also took part.

The results of the ANT supported the first hypothesis: programmers recorded faster global RTs than monolingual controls, and this difference was significant. Both adolescent and young adult programmers' groups recorded faster global RTs than both of the control groups. The young adult professionals recorded the fastest RTs, while the young adult controls recorded the slowest RTs; higher levels of experience and expertise were associated with greater executive control advantages over age-matched controls. Contrary to the second hypothesis, there was no significant difference in conflict effect. The results suggest that computer programmers may benefit from greater efficiencies in conflict monitoring network, the same mechanism thought to be responsible for the bilingual advantage.

The results of the Stroop test are more difficult to interpret. Contrary to predictions and to the ANT results, young adult programmers recorded the slowest response times; the difference was significant in the congruent condition. Largely as a result of this slow response in congruent conditions, the young adult programmers' group recorded a positive Stroop effect; they responded more quickly in incongruent than congruent conditions. A significantly slower global RT for programmers does not fit with the predictions of the study. One study from the bilingual literature is relevant: using the (admittedly quite different) spatial Stroop test, Bialystok, Craik and Luk (2008) found that bilinguals outperformed monolinguals on interference effect but not on global RT. This was the result of bilinguals responding 50ms more slowly than monolinguals on congruent trials (they were just 10ms faster than monolinguals on incongruent trials). So, whilst the slow reaction times by programmers in the congruent condition were not predicted, they are not entirely at odds with findings relating to the bilingual advantage. It should also be noted that in a neuroimaging study, Waldie, Badzakova-Trajkov, Milivojevic, and Kirk (2009) failed to find a significant difference in the performance of young adult monolinguals and bilinguals in a very similar Stroop colour-word task; however, they did find evidence that different areas of the brain were active during this task.

Somewhat surprisingly, three of the groups – both adolescent and young adult controls as well as the young adult programmers – were negatively affected by the presence of helpful cues (the word spelling out the correct answer) compared to neutral conditions; only the adolescent programmers made efficient use of these cues – they also showed less conflict cost than both control groups. It could be argued there are differences in the way computer programmers process text-based versus symbolic information – perhaps the Stroop activity is more like bug-fixing, and therefore experts access some automatic process to spot mistakes, but have no such practiced skill in utilizing useful cues – and that this difference only emerges following extensive practice, hence the contrast with the less experienced group. However, the unexpected pattern of these results, coupled with observed weaknesses in the experiment as the result of random presentation of the stimulus, raise the strong possibility that something other than the Stroop effect may have influenced the results. The results should therefore be interpreted with caution.

Overall, the present study has put forward a strong argument for the possibility that the “bilingual advantage” in executive control may also be found in computer programmers. The results of the study provide some evidence in support of that argument – there was a statistically significant advantage for the programming group in the ANT. However, the results of this small study are far from conclusive, and further research is needed before clear evidence of an association between computer programming and executive control can be claimed. The annual Young Rewired State event, which occurs in August, offers the opportunity to access large numbers of young coders; over 500 under 19s participated this year. There are also ample opportunities for prospective studies, for example, a new computer science GCSE will be available in schools from September and, in the younger age group, the organization Code Club aims to run computer programming courses in half of the UK’s primary schools.

It should be noted that the due to the quasi-experimental design, we cannot be certain that programming experience caused the differences in performance on the tasks; it may be that people with better executive control are more likely to persist with the development of programming language expertise because this advantage eases the learning process. A prospective study could help to clarify the direction of causation. However, if programming language acquisition is a benefit, rather than a cause, of enhanced executive control, this would not make the association any less interesting; in fact, it may make it more so. Learning to program is notoriously difficult – it is estimated that between 30% and 60% of university students fail the first programming course (Dehnadi & Bornat, 2006) and predicting which students are likely to make the grade has proved equally difficult (Bornat, Dehnadi, & Simon, 2008). If other studies replicate an association between executive control and computer programming, it would be interesting to compare the performance of bilinguals and monolinguals on introductory programming courses. It has been argued that since most bilinguals acquire two languages as a result of circumstance, rather than as a result of social advantages or cognitive abilities, the bilingual advantage is likely to be causal (though this argument is not yet been supported by prospective studies). If second language acquisition could help to develop the cognitive prerequisites of learning to program, this could have important implications for computer science education.

[Words: 11,341, excluding references and diagrams]

# References

Ben-Zeev, S. (1977). The Influence of Bilingualism on Cognitive Strategy and Cognitive Development. *Child Development, 48*(3), 1009–1018.

Bergersen, G. R., & Gustafsson, J. E. (2011). Programming skill, knowledge and working memory among professional software developers from an investment theory perspective. *Journal of Individual Differences, 32*(4), 201–209.

Bialystok, E. (2005). Consequences of bilingualism for cognitive development. In J. F. Kroll & A. M. B. De Groot (Eds.), *Handbook of bilingualism: psycholinguistic approaches* (pp. 417–453). New York, NY: Oxford University Press.

Bialystok, E. (2006). Effect of Bilingualism and Computer Video Game Experience on the Simon Task. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale, 60*(1), 68-79.

Bialystok, E. (2008). Bilingualism: The good, the bad, and the indifferent. *Bilingualism: Language and Cognition, 12*(01), 3.

Bialystok, E., Craik, F., & Luk, G. (2008). Cognitive control and lexical access in younger and older bilinguals. *J Exp Psychol Learn Mem Cogn, 34*(4), 859-873.

Bialystok, E., Craik, F. I., & Freedman, M. (2007). Bilingualism as a protection against the onset of symptoms of dementia. *Neuropsychologia, 45*(2), 459-464.

Bialystok, E., Craik, F. I., Klein, R., & Viswanathan, M. (2004). Bilingualism, aging, and cognitive control: evidence from the Simon task. *Psychol Aging, 19*(2), 290-303.

Bialystok, E., Craik, F. I. M., Green, D. W., & Gollan, T. H. (2009). Bilingual Minds. *Psychological Science in the Public Interest, 10*(3), 89-129.

Bialystok, E., Craik, F. I. M., & Luk, G. (in press). Bilingualism: consequences for mind and brain. *Trends in Cognitive Sciences*.

Bialystok, E., & DePape, A. M. (2009). Musical expertise, bilingualism, and executive functioning. *J Exp Psychol Hum Percept Perform, 35*(2), 565-574.

Bialystok, E., Martin, M. M., & Viswanathan, M. (2005). Bilingualism across the lifespan: The rise and fall of inhibitory control. *International Journal of Bilingualism, 9*(1), 103-119.

Bornat, R., Dehnadi, S., & Simon. (2008). *Mental models, Consistency and Programming Aptitude.* Paper presented at the Proceedings of the tenth conference on Australasian computing education, Wollongong, NSW, Australia.

Boroditsky, L. (2001). Does language shape thought? Mandarin and English speakers' conceptions of time. [Research Support, U.S. Gov't, Non-P.H.S.]. *Cogn Psychol, 43*(1), 1-22. doi: 10.1006/cogp.2001.0748

Botvinick, M. M., Cohen, J. D., & Carter, C. S. (2004). Conflict monitoring and anterior cingulate cortex: an update. *Trends Cogn Sci, 8*(12), 539-546.

Brefczynski-Lewis, J. A., Lutz, A., Schaefer, H. S., Levinson, D. B., & Davidson, R. J. (2007). Neural correlates of attentional expertise in long-term meditation practitioners. *Proc. Nat. Acad. Sci., 104*(27), 11483–11488.

Carlson, S. M., & Meltzoff, A. N. (2008). Bilingual experience and executive functioning in young children. *Dev Sci, 11*(2), 282-298.

Cherubini, M., Venolia, G., DeLine, R., & Ko, A. J. (2007). *Let's go to the whiteboard: how and why software developers use drawings.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose, California, USA.

Cohen, A., & Haberman, B. (2007). Computer Science: A Language of Technology. *Inroads — SIGCSE Bulletin, 39*(4).

Colzato, L. S., Bajo, M. T., van den Wildenberg, W., Paolieri, D., Nieuwenhuis, S., La Heij, W., & Hommel, B. (2008). How does bilingualism improve executive control? A comparison of active and reactive inhibition mechanisms. *J Exp Psychol Learn Mem Cogn, 34*(2), 302-312.

Costa, A., Hernandez, M., Costa-Faidella, J., & Sebastian-Galles, N. (2009). On the bilingual advantage in conflict processing: now you see it, now you don't. *Cognition, 113*(2), 135-149.

Costa, A., Hernandez, M., & Sebastian-Galles, N. (2008). Bilingualism aids conflict resolution: evidence from the ANT task. *Cognition, 106*(1), 59-86.

Crystal, D. (1997). *English as a global language*. Cambridge, UK: Cambridge University Press.

Dehnadi, S., & Bornat, R. (2006). *The camel has two humps (working title)*. School of Computing, Middlesex University, UK.

Détienne, F. (2002). *Software design: cognitive aspects*. New York, NY, USA.: Springer-Verlag New York, Inc.

Emmorey, K., Luk, G., Pyers, J. E., & Bialystok, e. (2008). The Source of Enhanced Cognitive Control in Bilinguals: Evidence From Bimodal Bilinguals. *Psychological Science in the Public Interest, 19*(12), 1201-1206.

Fan, J., McCandliss, B. D., Sommer, T., Raz, A., & Posner, M. I. (2002). Testing the Efficiency and Independence of Attentional Networks. *Journal of Cognitive Neuroscience, 14*(3), 340–347.

Green, D. W. (1998). Mental control of the bilingual lexico-semantic system. *Bilingualism: Language and Cognition, 1*, 67-81

Grosjean, F. (1992). Another view of bilingualism. In R. Harris (Ed.), *Cognitive Processing in Bilinguals*. Amsterdam: North Holland.

Hernandez, A. E., Dapretto, M., Mazziotta, J., & Bookheimer, S. (2001). Language switching and language representation in Spanish-English bilinguals: an fMRI study. *Neuroimage, 14*(2), 510-520.

Hernandez, A. E., Li, P., & MacWhinney, B. (2005). The emergence of competing modules in bilingualism. *Trends Cogn. Sci. , 9*(5), 220–225.

Hilchey, M. D., & Klein, R. M. (2011). Are there bilingual advantages on nonlinguistic interference tasks? Implications for the plasticity of executive control processes. *Psychon Bull Rev, 18*(4), 625-658.

Hockett, C. F. (1960). The origin of speech. *Scientific American, 203*, 88-96.

Huttenlocher, P. R. (2002). *Neural plasticity: The effects of environment on the development of the cerebral cortex*. Cambridge, MA: Harvard University Press.

Ianco-Worrall, A. D. (1972). Bilingualism and Cognitive Development. *Child Development, 43*(4), 1390-1400.

Jarvis, S. (2009). Lexical transfer. In A. Pavlenko (Ed.), *The bilingual mental lexicon: interdisciplinary approaches*. Clevedon, UK: Multilingual Matters.

Kessler, C., & Quinn, M. E. (1980). Positive effects of bilingualism on science problem-solving abilities. In J. E. Alatis (Ed.), *Current issues in bilingual education: Proceedings of the Georgetown Roundtable on Languages and Linguistics*. Washington, DC: Georgetown University Press.

Kessler, C., & Quinn, M. E. (1987). Language minority children's linguistic and cognitive creativity. *Journal of Multilingual and Multicultural Development, 8*, 173–186.

Kokol, P., Podgorelec, V., Zorman, M., Kokol, T., & Njivar, T. (1999). Computer and Natural Language Texts - A Comparison Based on Long-Range Correlations. *Journal of the American Society for Information Science, 50*(14), 1295-1301.

Kuhl, P. K. (2004). Early language acquisition: cracking the speech code. *Nat Rev Neurosci, 5*(11), 831-843.

Lehrer, R., & DeBernard, A. (1987). Language of learning and language of computing: The perceptual-language model. *Journal of Educational Psychology, 79*(1), 41–48.

Liao, Y. K. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research, 7*(3), 251–268.

Luk, G., Anderson, J. A., Craik, F. I., Grady, C., & Bialystok, E. (2010). Distinct neural correlates for two types of inhibition in bilinguals: response inhibition versus interference suppression. *Brain Cogn, 74*(3), 347-357.

Martin-Rhee, M. M., & Bialystok, E. (2008). The development of two types of inhibitory control in monolingual and bilingual children. *Bilingualism: Language and Cognition, 11*(01).

Meisel, J. (1989). Early differentialation of languages in bilingual children. In K. Hyltenstam & L. Obler (Eds.), *Bilingualism across the lifespan. Aspects of aquisition, maturity and loss.* Cambridge, UK.: Cambridge University Press.

Murnane, J. (2006). *Programming Languages for Beginners: A Comparison of Textual and Graphic Programming Environments for Novice Programmers.* University of Melbourne, Melbourne.

Murnane, J. S. (1993). The psychology of computer languages for introductory programming courses. *New ideas in psychology, 11*(2), 213–228.

Norman, K. L. (2008). *Cyberpsychology: An Introduction to Human-Computer Interaction*. New York: Cambridge University Press

Ormerod, T. (1990). Human Cognition and Programming. In J. M. Hoc, T. R. G. Green, R. Samurçay & D. J. Gilmore (Eds.), *Psychology of Programming.* (pp. 63–82). London: The European Association of Cognitive Ergonomics and Academic Press.

Palumbo, D. B. (1990). Programming Language/Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research, 60*(1), 65-89.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY: Basic Books.

Pascual-Leone, A. (2001). The Brain That Plays Music and Is Changed by It. *Annals of the New York Academy of Sciences, 930: 315–329., 930*, 315–329.

Pavlenko, A., & Jarvis, S. (2002). Bidirectional Transfer. *Applied Linguistics, 23*(2), 190–214.

Pea, R. D. (1983). Logo Programming and Problem Solving. [Technical Report No. 12.]. New York: Center for Children and Technology Bank Street College of Education.

Pea, R. D., & Kurland, D. M. (1984). On the Cognitive Effects of Learning Computer Programming. *New Ideas Psychol, 2*, 137–168.

Peal, E., & Lambert, W. (1962). The relation of bilingualism to intelligence. *Psychological Monographs, 76*(546), 1–23.

Peppler, K. A., & Warschauer, M. (2011). Uncovering Literacies, Disrupting Stereotypes: Examining the (Dis)Abilities of a Child Learning to Computer Program and Read. *International Journal of Learning and Media, 3*(3), 15-41.

Philipp, A. M., & Koch, I. (2009). Inhibition in language switching: what is inhibited when switching between languages in naming tasks? *J Exp Psychol Learn Mem Cogn, 35*(5), 1187-1195.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., . . . Silver, J. (2009). Scratch. *Communications of the ACM, 52*(11), 60.

Ricciardelli, L. A. (1992). Bilingualism and cognitive development in relation to threshold theory. *Journal of Psycholinguistic Research, 21*, 301–316.

Sleeman, D., Putnam, R. T., Baxter, J., & Kuspa, L. (1986). Pascal and High-School Students: A study of errors. *Journal of Educational Computing Research. Special issue on Novice Programming*, 5–23.

Soloway, E., & Spohrer, J. (1989). *Studying the Novice Programmer*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Spelke, E. S. (2003). What makes us smart? Core knowledge and natural language. In D. Gentner & S. Goldin-Meadow (Eds.), *Language in Mind: Advances in the Investigation of Language and Thought*. Cambridge, MA: MIT Press.

Stern, Y. (2002). What is cognitive reserve? Theory and research application of the reserve concept. *J. Int. Neuropsychol. Soc., 8*, 448–460.

Tao, L., Marzecova, A., Taft, M., Asanowicz, D., & Wodniecka, Z. (2011). The efficiency of attentional networks in early and late bilinguals: the role of age of acquisition. *Front Psychol, 2*, 123.

Thierry, G., & Wu, Y. J. (2007). Brain potentials reveal unconscious translation during foreign-language comprehension. *Proc Natl Acad Sci U S A, 104*(30), 12530-12535.

Tremblay, G. P., & Sorenson, P. G. (1986). *The Theory and Practice of Compiler Writing*. New York: McGraw-Hill.

Waldie, K., Badzakova-Trajkov, G., Milivojevic, B., & Kirk, I. (2009). Neural activity during Stroop colour-word task performance in late proficient bilinguals: a functional Magnetic Resonance Imaging study. *Psychology and Neuroscience, 2*(2), 125-136.

Wechsler, D. (1999). *Wechsler Abbreviated Scale of Intelligence (WASI)*. San Antonio, TX: Harcourt Assessment.

Weinberg, G. M. (1971). *The Psychology of computer programming*: Van Nostrand Reinhold.

Witschital, P. (1995). TRAPS – An intelligent tutoring environment for novice programmers. In K. F. Wender, F. Schmalhofer & H.-D. Böcker (Eds.), *Cognition and computer programming*. Norwood, New Jersey: Ablex Publishing Corporation.

Woollett, K., & Maguire, E. A. (2011). Acquiring "the Knowledge" of London's Layout Drives Structural Brain Changes. *Current Biology, 21*(24), 2109-2114.

Yelland, G. W., Pollard, J., & Mercuri, A. (1993). The metalinguistic benefits of limited contact with a second language. *Applied Psycholinguistics, 14*, 423-444.