Comprehension of computer code relies primarily on domain-general executive resources

Anna A. Ivanova[1,2], Shashank Srikant[3], Yotaro Sueoka[1], Hope H. Kean[1,2], Riva Dhamala[4], Una-May O'Reilly[3], Marina U. Bers[4], Evelina Fedorenko[1,2,5]

[1]*Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology*

[2]*McGovern Institute for Brain Research, Massachusetts Institute of Technology*

[3]*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology*

[4]*Eliot-Pearson Department of Child Study and Human Development, Tufts University*

[5]*Department of Psychiatry, Massachusetts General Hospital*

Abstract

Computer programming is a novel cognitive tool that has transformed modern society. An integral part of programming is code comprehension: the ability to process individual program tokens, combine them into statements, which, in turn, combine to form a program. What cognitive and neural mechanisms support this ability to process computer code? Here, we used fMRI to investigate the role of two candidate brain systems in code comprehension: the multiple demand (MD) system, typically recruited for math, logic, problem solving, and executive function, and the language system, typically recruited for linguistic processing. Across two experiments, we examined brain responses to code written in two programing languages: Python, a text-based programming language (Experiment 1) and ScratchJr, a graphical programming language for children (Experiment 2). To isolate neural activity evoked by code comprehension per se rather than by processing program content, we contrasted responses to code problems with responses to content-matched sentence problems. We found that the MD system exhibited strong bilateral responses to code in both experiments. In contrast, the language system responded strongly to sentence problems, but only weakly or not at all to code problems. We conclude that code comprehension relies primarily on domain-general executive resources, demonstrating that the MD system supports the use of novel cognitive tools even when the input is structurally similar to natural language.

Corresponding author: Anna Ivanova (annaiv@mit.edu)

Date: April 2020

Introduction

The human mind is endowed with a remarkable ability to support novel cognitive skills, such as reading, writing, map-based navigation, mathematical reasoning, and scientific logic. Recently, humanity has invented another powerful cognitive tool: computer programming. The ability to flexibly instruct programmable machines has led to a rapid technological transformation of communities across the world (Ensmenger, 2012); however, little is known about the cognitive and neural systems that underlie computer programming skills.

Here, we investigate which neural systems support one critical aspect of computer programming: computer code comprehension. By code comprehension, we refer to a set of cognitive processes that allow programmers to interpret individual program tokens (such as keywords, variables, and function names), combine them to extract the meaning of program statements, and, finally, combine the statements into a mental representation of the entire program. It is important to note that code comprehension may be cognitively and neurally separable from cognitive operations required to process program content, i.e. the actual operations described by code. For instance, to predict the output of the program that sums the first three elements of an array, the programmer should identify the relevant elements and then mentally perform the summation. Most of the time, processing program content recruits a range of cognitive processes known as computational thinking (Wing, 2006, 2011), which include decomposing a problem into sub-problems, pattern identification, pattern generalization/abstraction, algorithm design, and recursive reasoning (e.g., Kao, 2011). These cognitive operations are notably different from code comprehension per se and may not require programming knowledge at all (Guzdial, 2008). Thus, any research study where people read computer programs should account for two separate cognitive phenomena: processing computer code that comprises the program (i.e., code comprehension) and mentally simulating the procedures described in the program (i.e., processing problem content).

Given that code comprehension is a novel cognitive tool, typically acquired in late childhood or in adulthood, we expect it to draw on preexisting cognitive systems. However, the question of which cognitive processes support code comprehension is non-trivial. Unlike some cognitive inventions that are primarily linked to a single preexisting system (e.g., reading/writing building on spoken language), code comprehension bears parallels to multiple cognitive domains. Like any cognitively demanding activity, it may rely on domain-general executive functions, including working memory and cognitive control (Bergersen & Gustafsson, 2011; Nakagawa et al., 2014; Nakamura et al., 2003). In addition, it may draw on the cognitive systems associated with math and logic (McNamara, 1967; Papert, 1972), in line with the traditional construal of coding as problem-solving (Dalbey & Linn, 1985; Ormerod, 1990; Pea & Kurland, 1984; Pennington & Grabowski, 1990). Finally, code comprehension may rely on the system that supports comprehension of natural languages: to successfully process both natural and computer languages, we need to access stored meanings of words/tokens and combine them using hierarchical syntactic rules (Fedorenko et al., 2019; Murnane, 1993; Papert, 1993) — a similarity that, in theory, should make the language circuits well-suited for processing computer code.

Neuroimaging research is well positioned to disentangle the relationship between code comprehension and other cognitive domains. Many cognitive processes are known to evoke activity in specific brain networks: thus, observing activity for the task of interest in a network with a known function can indicate which cognitive processes are likely involved in that task (Mather et al., 2013). Prior research (Duncan, 2010, 2013; Duncan & Owen, 2000) has shown that executive processes — such as attention, working memory, and cognitive control — recruit a set of bilateral frontal and parietal brain regions collectively known as the multiple demand (MD) system. If code comprehension primarily relies on domain-general executive processes, we expect to observe code-evoked responses within the MD system, distributed across both hemispheres. Math and logic also evoke responses within the MD system (Fedorenko et al., 2013), although

this activity tends to be left-lateralized (Goel & Dolan, 2001; Micheloyannis et al., 2005; Monti et al., 2007, 2009; Pinel & Dehaene, 2009; Prabhakaran et al., 1997; Reverberi et al., 2009). If code comprehension draws on the same mechanisms as math and logic, we expect to observe left-lateralized activity within the MD system. Finally, comprehension of natural language recruits a set of left frontal and temporal brain regions known as the language system (e.g., Fedorenko & Thompson-Schill, 2014). These regions respond robustly to linguistic input (visual or auditory; Bemis & Pylkkänen, 2013; Deniz et al., 2019; Fedorenko et al., 2010) but show little or no response to tasks in non-linguistic domains, such as executive functions, math, logic, music, action observation, or non-linguistic communicative signals, such as gestures (Fedorenko et al., 2011; Jouravlev et al., 2019; Monti et al., 2009, 2012; Pritchett et al., 2018; see Fedorenko & Blank, 2020, for a review). If code comprehension relies on the same circuits that map form to meaning in natural language, we expect to see activity within the language system.

Evidence from prior neuroimaging investigations of code comprehension is inconclusive: some studies have reported left-lateralized activity in the regions that roughly correspond to the language system (Siegmund et al., 2014, 2017), whereas others have observed activity in frontal and parietal regions resembling the MD system (Floyd et al., 2017; Huang et al., 2019). However, none of these prior studies sought to explicitly distinguish code comprehension from other programming-related processes, and most of them did not report whether the code-responsive brain regions they isolated were active during non-programming tasks (such as working memory, math, or language; cf. Huang et al, 2019).

Here, we use functional magnetic resonance imaging (fMRI) to evaluate the role of the MD system and the language system in computer code comprehension. Three design features that were lacking in earlier neuroimaging studies of programming allow us to evaluate the relative contributions of these two candidate systems. First, we contrast neural responses evoked by code problems with those evoked by content-matched sentence problems (**Figure 1A**); this comparison allows us to disentangle activity evoked by code comprehension from activity evoked by the underlying program content (which is matched across code and sentence problems).

Second, we use independent 'localizer' tasks (Brett et al., 2002; Fedorenko et al., 2010; Saxe et al., 2006) to identify our networks of interest: a working memory task to localize the MD system and a passive reading task to localize the language system (**Figure 1B)**. The functional localization approach obviates the reliance on the much-criticized 'reverse inference' reasoning (Poldrack, 2006, 2011), whereby functions are inferred from coarse macro-anatomical landmarks. Instead, we can directly interpret code-evoked activity within functionally defined regions of interest (Mather et al., 2013). In addition, localization of the MD and language networks is performed in individual participants, which is important given substantial variability in their precise locations across individuals (Fedorenko & Blank, 2020; Shashidhara, Spronkers, et al., 2019) and leads to higher sensitivity and functional resolution (Nieto-Castañón & Fedorenko, 2012).

Third, to draw general conclusions about code comprehension, we investigate two very different programming languages: Python, a popular general-purpose programming language, and ScratchJr, an introductory visual programming language for creating animations designed for young children (Bers & Resnick, 2015). In the Python experiment, we further examine two problem types (math problems and string manipulation) and three basic types of program structure (sequential statements, *for* loops, and *if* statements). Comprehension of both Python and ScratchJr code requires retrieving the meaning of program tokens and combining them into statements, despite the fact that the visual features of the tokens in the two languages are very different (text vs. images). If a brain system is involved in code comprehension, we expect its response to generalize across programming languages and problem types, similar to how distinct natural languages in bilinguals and multilinguals draw on the same language regions (Kroll et al., 2015).

Taken together, the design features of our study allow us to draw precise and generalizable conclusions about the neural basis of code comprehension.

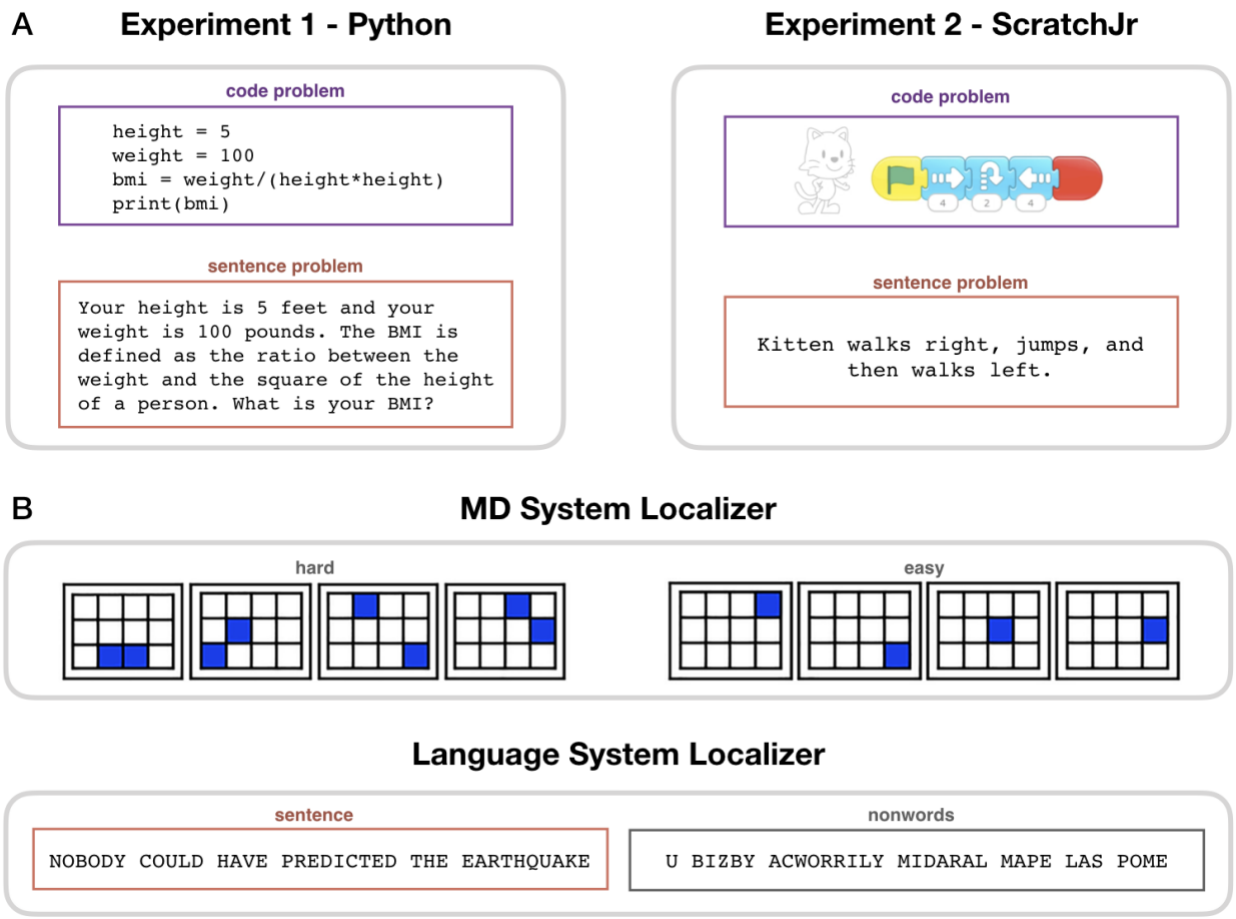

Figure 1. *Experimental paradigms. (**A**) Main task. During code problem trials, participants were presented with snippets of code in Python (Experiment 1) or ScratchJr (Experiment 2); during sentence problem trials, they were presented with text problems that were matched in content with the code stimuli. No participants saw both the code and the sentence versions of each problem. (**B**) Localizer tasks. The MD localizer (top) included a hard condition (memorizing positions of 8 blocks appearing two at a time) and an easy condition (memorizing positions of 4 blocks appearing one at a time). The language localizer (bottom) included a sentence reading and a nonword reading condition, with the words/nonwords appearing one at a time.*

Results

Participants performed a program comprehension task inside an MRI scanner. In each trial, participants, all proficient in the target programming language, read either a code problem or a content-matched sentence problem (**Figure 1A**) and were asked to predict the output. In Experiment 1 (24 participants, 15 women), code problems were written in Python, a general-purpose text-based programming language (Sanner, 1999). In Experiment 2 (19 participants, 12 women), code problems were written in ScratchJr, an introductory graphical programming language developed for kids aged 5-7 (Bers, 2018). Both experiments were conducted with adults to facilitate result comparison. Good behavioral performance confirmed that participants were proficient in the relevant programming language and engaged with the task (Python: 99.6% response rate, 85% accuracy on code problems; ScratchJr: 98.6% response rate, 79% accuracy on code problems; see **Supplementary Figure 1** for detailed behavioral results). Participants additionally performed two functional localizer tasks: a hard vs. easy spatial working memory task, used to define the

MD system, and a sentence vs. nonword reading task, used to define the language system (**Figure 1B**; see **Methods** for details).

We then contrasted neural activity in the MD and language systems during code problem comprehension with activity during (a) sentence problem comprehension and (b) the nonword reading condition from the language localizer task. Sentence problem comprehension requires simulating the same operations as code problem comprehension (mathematical operations or string manipulation for Python, video simulation for ScratchJr), so contrasting code problems with sentence problems allows us to isolate neural responses evoked by code comprehension from responses evoked by processing problem content. Nonword reading elicits weak responses in both the language system and the MD system (in the language system, this response likely reflects low-level perceptual and/or phonological processing; in the MD system, it likely reflects the basic task demands associated with reading). Because the nonword response is much weaker than responses to the localizer conditions of interest (Fedorenko et al., 2010; Mineroff et al., 2018), nonword reading can serve as a control condition for both the MD and language systems, providing a more stringent baseline than simple fixation. Given the abundant evidence that the MD system and the language system are each strongly functionally interconnected (Blank et al., 2014; Mineroff et al., 2018), we perform the key analyses at the system level.

MD system exhibits robust and generalizable bilateral responses during code comprehension

We found strong bilateral responses to code problems within the MD system in both Experiments 1 and 2 (**Figure 2**). These responses were stronger than responses to both the sentence problem condition (Python: $\beta = 0.98$, $p < 0.001$, ScratchJr: $\beta = 1.10$, $p < 0.001$) and the control nonword reading condition (Python: $\beta = 2.53$, $p < 0.001$; ScratchJr: $\beta = 1.20$, $p < 0.001$). The fact that code problems drove the MD system more strongly than content-matched sentence problems (despite the fact that sentence problems generally took longer to respond to; see **Supplementary Figure 1**) demonstrate that the MD system responds to code comprehension specifically rather than simply being activated by the underlying problem content.

To further test the generalizability of MD responses, we capitalized on the fact that our Python stimuli systematically varied along two dimensions: (1) problem type (math problems vs. string manipulation), and (2) problem structure (sequential statements, *for* loops, *if* statements). Strong responses were observed in the MD system (**Figure 3A**) regardless of problem type ($\beta = 3.03$, $p < 0.001$; no difference between problem types) and problem structure ($\beta = 3.14$, $p < 0.001$; sequential problems evoked a slightly weaker response, $\beta = -0.19$, $p = 0.002$). This analysis demonstrates that the responses were not driven by one particular type of problem or by mental operations related to the processing of a particular code structure.

We also tested whether MD responses to code showed a hemispheric bias similar to what is typically seen for math and logic problems (Goel & Dolan, 2001; Micheloyannis et al., 2005; Monti et al., 2007, 2009; Pinel & Dehaene, 2009; Prabhakaran et al., 1997; Reverberi et al., 2009). Neither Python nor ScratchJr problems showed a left-hemisphere bias for code comprehension. For Python, the size of the code problems>sentence problems effect did not interact with hemisphere ($\beta = 0.10$, *n.s.*), even though the magnitude of responses to code problems as compared to nonword reading was stronger in the left hemisphere ($\beta = 0.68$, $p < 0.001$). These results show that neural activity evoked by Python code comprehension was bilaterally distributed but that activity evoked by the underlying problem content was left-lateralized. For ScratchJr, the size of the code problems>sentence problems effect interacted with hemisphere, with stronger responses in the right hemisphere ($\beta = 0.57$, $p = 0.001$), perhaps reflecting the bias of the right hemisphere toward visuo-spatial processing (Corballis, 2003; Hugdahl, 2011; Sheremata et al., 2010).

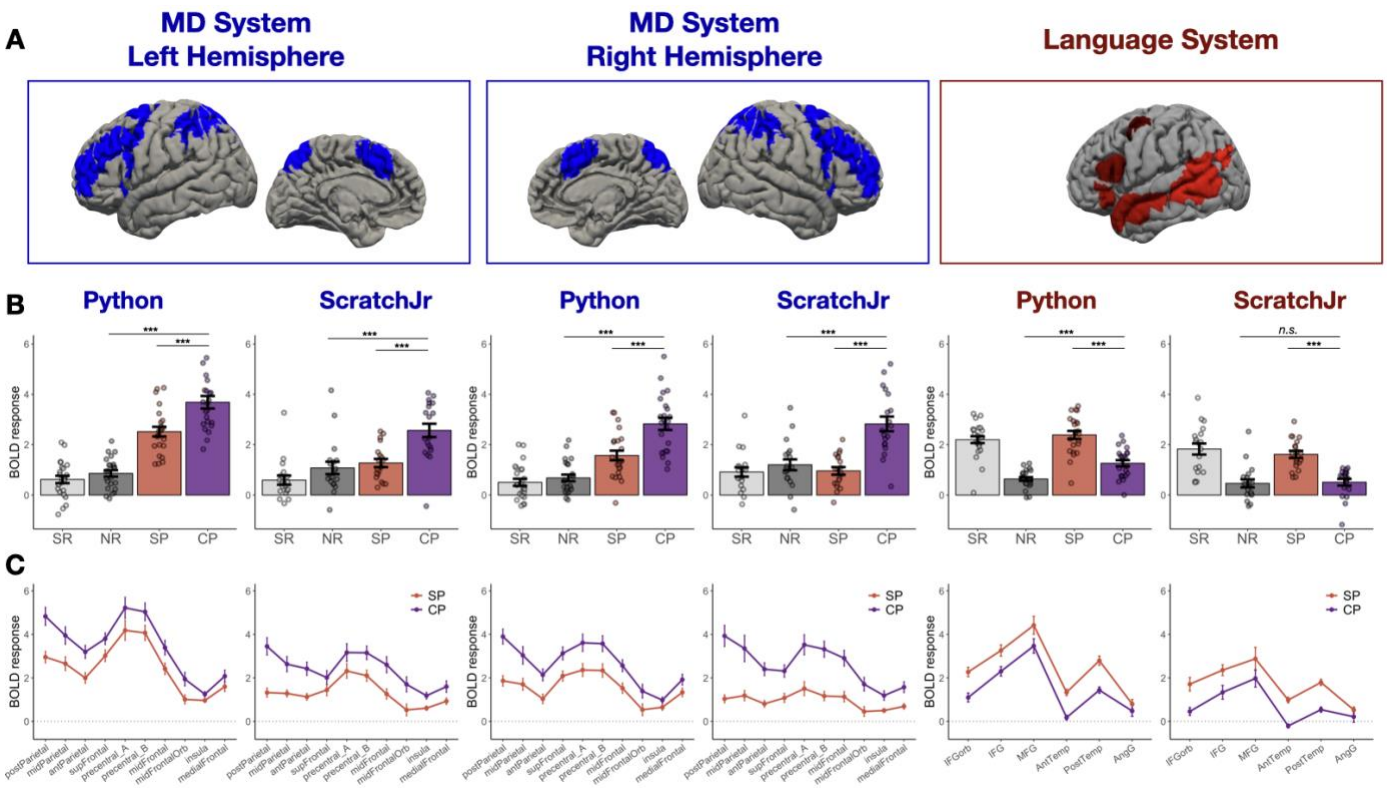

Figure 2. *(**A**) Candidate brain systems of interest. The areas shown represent the "parcels" used to define the MD and language systems in individual participants (see **Methods** and Figure 2). (**B**) Mean responses to the language localizer conditions (SR - sentence reading and NR - nonwords reading) and to the critical task (SP - sentence problems and CP - code problems) in systems of interest across programming languages (Python and ScratchJr). In the MD system, we see strong responses to code problems in both hemispheres and to both programming languages; the fact that this response is stronger than the response to content-matched sentence problems suggests that it reflects activity evoked by code comprehension per se rather than just activity evoked by problem content. In the language system, responses to code problems elicit a response that is substantially weaker than that elicited by sentence problems; further, only in Experiment 1 do we observe responses to code problems that are reliably stronger than the responses to the language localizer control condition (nonword reading). (**C**) Responses to sentence problems (SP) and code problems (CP) broken down by region of interest. Abbreviations: mid – middle, ant – anterior, post – posterior, orb – orbital, MFG – middle frontal gyrus, IFG – inferior frontal gyrus, temp – temporal lobe, AngG – angular gyrus. Here and elsewhere, error bars show standard error of the mean.*

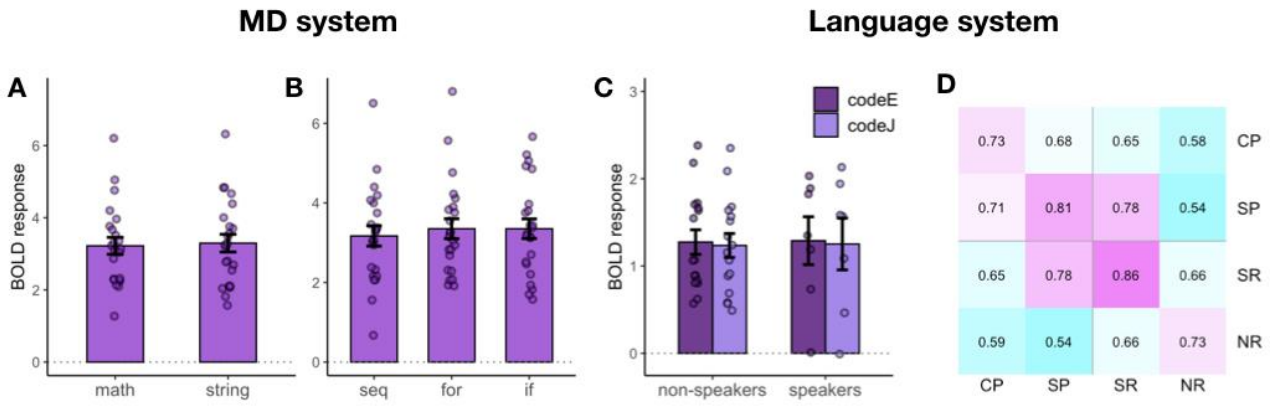

Figure 3*. Follow-up analyses of brain responses to Python code. (**A**) MD system responses to math problems vs. string manipulation problems. (**B**) MD system responses to code with different structure (sequential vs. for loops vs. if statements). (**C**) Language system responses to code problems with English identifiers (codeE) and code problems with Japanese identifiers (codeJ) in participants with no knowledge of Japanese (non-speakers) and some knowledge of Japanese (speakers). (**D**) Spatial correlation analysis of voxel responses within the language system during the main experiment (CP – code problems and SP – sentence problems) with the language localizer conditions (SR – sentence reading and NR – nonword reading). Each cell shows a correlation between voxel-level activation patterns for each condition. Within-condition similarity is estimated by correlating activation patterns across independent runs.*

Individual fROI analysis (see **Supplementary Figure 2** for fROI parcel locations) demonstrated that all MD fROIs except those located in left and right insula responded significantly more strongly to Python code problems compared to sentence problems (**Figure 2C**; see **Supplementary Table 1** for all fROI statistics). Responses to Scratch Jr were significantly stronger than responses to sentence problems in 6/10 left hemisphere fROIs (except the ones in superior frontal gyrus, the dorsal part of precentral gyrus, the insula, and the medial frontal fROI) and 8/10 right hemisphere fROIs (except the ones in the insula and in the medial frontal gyrus; **Figure 2C**; see **Supplementary Table 2** for all fROI statistics). These analyses demonstrate that code processing is broadly distributed across the MD system rather than being localized to a particular region or to a small subset of regions.

Overall, we show that MD responses to code are strong, do not exclusively reflect responses to problem content, are observed across most MD fROIs, and generalize across programming languages and problem types.

No MD fROIs are selective for code comprehension

To determine whether any fROIs were driven selectively (preferentially) by code problems relative to other cognitively demanding tasks, we contrasted individual fROI responses to code problems with responses to a hard working memory task from the MD localizer experiment. Three fROIs located in the left frontal lobe ("precentral_A", "precentral_B", and "midfrontal") exhibited stronger responses to Python code problems than to the hard working memory task ($\beta = 1.17$, $p = 0.001$; $\beta = 2.03$, $p < 0.001$; and $\beta = 0.86$, $p = 0.009$, respectively; **Supplementary Figure 3**). However, the magnitude of the code problems > sentence problems contrast in these regions ($\beta = 1.04, 0.97, 0.97$) was comparable to the average response magnitude across all MD fROIs (average $\beta = 1.03$), suggesting that the high response was caused by processing the underlying problem content rather than by code comprehension per se. Furthermore, neither these nor any other MD fROIs exhibited higher responses to ScratchJr code compared to the hard working memory task (in fact, the fROI in "precentral_A" did not even show a significant code problems>sentence problems

response). We conclude that code comprehension is broadly supported by the MD system (similarly to, e.g., intuitive physical inference; Fischer et al., 2016), but no MD regions are functionally specialized to process computer code.

Language system responses during code comprehension are weak and inconsistent

The responses to code problems within the language system (**Figure 2**) were weaker than responses to sentence problems in both experiments (Python: $\beta = 1.00$, $p < 0.001$; ScratchJr: $\beta = 0.99$, $p < 0.001$). Furthermore, although the responses to code problems were stronger than the responses to nonword reading for Python ($\beta = 0.80$, $p < 0.001$), this was not the case for ScratchJr ($\beta = 0.15$, *n.s.*), suggesting that the language system is not consistently engaged during computer code comprehension.

We further tested whether responses to Python code problems within the language system may be driven by the presence of English words. Our stimuli were constructed such that half of the Python problems contained meaningful identifier names, and in the other half, the English identifiers were replaced with their Japanese translations, making them semantically meaningless for non-speakers of Japanese. For this analysis, we divided our participants into two groups — those with no reported knowledge of Japanese (N=18) and those with some knowledge of Japanese (N=6) — and compared responses within their language regions to code problems with English vs. Japanese identifiers (**Figure 3C**). We found no effect of identifier language ($\beta = 0.03$, *n.s.*), knowledge of Japanese ($\beta = 0.01$, *n.s.*), or interaction between them ($\beta = 0.09$, *n.s.*), indicating that the language system's response to Python code was not driven by the presence of semantically transparent identifiers. This result is somewhat surprising given the language system's strong sensitivity to word meanings (e.g., Anderson et al., 2019; Binder et al., 2009; Fedorenko et al., 2010, 2020; Pereira et al., 2018). One possible explanation is that participants do not deeply engage with the words' meanings in these problems because these meanings are irrelevant to finding the correct solution.

Finally, we investigated whether the observed responses to Python code problems were primarily driven by code comprehension specifically or rather by the underlying problem content. To do this, we compared the fine-grained patterns of activation within the language system evoked by code and sentence problems. We computed voxel-wise spatial correlations within and between responses to code problems and sentence problems, as well as correlations between responses to these conditions and to sentence/nonword reading conditions from the language localizer task (**Figure 3D**). We hypothesized that, if the system is driven by processes involved in code comprehension, the activation profiles for code and sentence problems will differ; in contrast, if the system is driven by problem content, the activation profiles for code and sentence problems will be similar. We found that the activation patterns were highly correlated between code and sentence problems ($r=0.70$). These correlation values were higher than the correlations between code problems and sentence reading (0.70 vs. 0.65; $p < 0.001$), although somewhat lower than the correlations within the code problem condition (0.70 vs. 0.73; $p < 0.001$). These results suggest that the language system's response to code is largely (although not completely) driven by problem content.

Overall, we found that the language system responded to code problems written in Python but not in ScratchJr. Furthermore, Python responses appeared to be largely driven by the processing of problem content rather than by code comprehension per se, leading us to conclude that the language system does not support code comprehension in proficient programmers.

No consistent evidence of code-responsive regions outside the MD/language systems

To search for code-responsive regions that might fall outside the MD and language systems, we performed a whole-brain GSS analysis (Fedorenko et al., 2010). GSS analysis serves the same goal as the traditional

random-effects voxel-wise analysis (Holmes & Friston, 1998) but accommodates inter-individual variability in the precise locations of functional regions, thus maximizing the likelihood of finding responsive regions. We searched for areas of activation for the code problems>sentence problems contrast (separately for Python and ScratchJr) that were spatially similar across participants. We then examined the response of such regions to code and sentence problems (using an across-runs cross-validation procedure; e.g., Nieto-Castañón & Fedorenko, 2012), as well as to conditions from the two localizer experiments. In both experiments, the discovered regions spatially resembled the MD system (**Supplementary Figures 4, 5**). For Python, any region that responded to code also responded to the spatial working memory task (the MD localizer). In case of ScratchJr, some fROIs responded more strongly to code problems than to the spatial working memory task; these fROIs were located in early visual areas/ventral visual stream and therefore likely responded to low-level visual properties of ScratchJr code (which includes colorful icons, objects, etc.). The traditional random-effects group analyses revealed a similar activation pattern (**Supplementary Figures 6, 7**). These whole-brain analyses demonstrate that the MD system responds robustly and consistently to computer code, recapitulating the results of the fROI-based analyses (**Figures 2B, C**) and show that fROI-based analyses did not miss any non-visual code-responsive or code-selective regions outside the boundaries of the MD system.

Effect of proficiency on MD and language responses

We conducted an exploratory analysis to check whether engagement of MD and/or language system in code comprehension varies with the level of programming expertise. We correlated responses within each system with independently obtained proficiency scores for Experiment 1 participants (see the paper's website for details) and with in-scanner accuracy scores for Experiment 2 participants. No correlations were significant (see **Supplementary Figure 8**). However, due to a relatively low number of participants (N=24 and N=19, respectively), these results should be interpreted with caution.

Discussion

The ability to interpret computer code is a remarkable cognitive skill that bears parallels to diverse cognitive domains, including general executive functions, math, logic, and language. The fact that coding can be learned in adulthood suggests that it may rely on existing cognitive systems. Here, we tested the role of two candidate neural systems in computer code comprehension: the domain-general multiple demand (MD) system (Duncan, 2010) that has been linked to diverse executive demands and implicated in both math and logic (e.g., Amalric & Dehaene, 2019; Goel, 2007; Monti et al., 2007, 2009), and the language-selective system (Fedorenko et al., 2011) that has been linked to lexical and combinatorial linguistic processes (e.g., Bautista & Wilson, 2016; Fedorenko et al., 2010, 2020; Fedorenko, Nieto-Castañon, et al., 2012; Keller et al., 2001; Mollica et al., 2020). We found robust bilateral responses to code problems within the MD system, a pattern that held across two very different programming languages (Python and ScratchJr), types or problems (math and string manipulation), and problem structure (sequential statements, *for* loops, and *if* statements). In contrast, responses in the language system were substantially lower than those elicited by the content-matched sentence problems and exceeded responses to the control condition (nonwords reading) only for one of the two programming languages tested.

Our work uniquely contributes to the study of computer programming in the mind and brain by addressing two core issues that made it difficult to interpret results from prior studies. First, we disentangle responses evoked by code comprehension from responses to problem content (which is often not code-specific) by contrasting code problems with content-matched sentence problems. Our findings suggest that earlier reports of left-lateralized code-evoked activity (Siegmund et al., 2014) may reflect processing program content rather than code comprehension per se. This distinction should also be considered when interpreting

results of other studies of programming effects on brain activity, such as debugging (Castelhano et al., 2019), variable tracking (Ikutani & Uwano, 2014; Nakagawa et al., 2014), use of semantic cues or program layout (Fakhoury et al., 2018; Siegmund et al., 2017), program generation (Krueger et al., 2020), and programming expertise (Ikutani et al., 2020).

Second, we analyze responses in brain areas that are functionally localized in individual participants, allowing for straightforward interpretation of the observed responses (Mather et al., 2013; Saxe et al., 2006). This approach stands in contrast to the traditional approach, whereby neural responses are averaged across participants on a voxel-by-voxel basis, and the resulting activation clusters are interpreted via flawed 'reverse inference' from anatomy (e.g., Poldrack, 2006, 2011). Functional localization is particularly important when analyzing responses in fronto-temporo-parietal associative cortices, which are known to be functionally heterogeneous and variable across individuals (Blank et al., 2017; Braga et al., 2019; Fedorenko & Kanwisher, 2009; Shashidhara, Spronkers, et al., 2019).

MD system's engagement reflects the use of domain-general resources

The fact that the MD system responds to code problems over and above content-matched sentence problems underscores the role of domain-general executive processes in code comprehension. Although cognitive processes underlying code interpretation bear parallels to logic and math tasks (Papert, 1972; Pennington & Grabowski, 1990; Perkins & Simmons, 1988) and to natural language comprehension/generation (Fedorenko et al., 2019; Hermans & Aldewereld, 2017), the neural activity we observe primarily resembles activity observed in response to domain-general executive tasks (Fedorenko et al., 2013). In particular, code comprehension elicits bilateral responses within the MD system, in contrast to math and logic tasks that tend to elicit left-lateralized responses within the MD system, and in contrast to language tasks that elicit responses in the spatially and functionally distinct language system.

The process of code comprehension includes retrieving code-related knowledge from memory and applying it to the problems at hand. This application of task-relevant knowledge plausibly requires attention, working memory, inhibitory control, planning, and general flexible reasoning—cognitive processes long linked to the MD system (Duncan, 2010, 2013; Duncan & Owen, 2000; Miller & Cohen, 2001) in both humans (Assem et al., 2019; Shashidhara, Mitchell, et al., 2019; Woolgar et al., 2018) and non-human primates (Freedman et al., 2001; Miller et al., 1996; Mitchell et al., 2016). A recent study (Huang et al., 2019) reported neural overlap between operations on programming data structures and a mental rotation task within brain regions whose topography resembles that of the MD system. In our study, all code-responsive brain regions outside the visual cortex also responded robustly during a spatial memory task (**Supplementary Figures 4, 5**), similarly to the results reported in Huang et al. (2019). However, the MD system is not specifically tuned to spatial reasoning (Duncan, 2010; Fedorenko et al., 2013; Michalka et al., 2015), so the overlap between code comprehension and spatial reasoning likely reflects the engagement of highly domain-general cognitive processes, like working memory and cognitive control, as opposed to cognitive processes specific to spatial reasoning tasks.

Furthermore, given that no regions outside of the MD system showed code-specific responses, it must be the case that code-specific knowledge representations are also *stored* within this system (see Hasson et al., 2015, for a general discussion of the lack of distinction between storage and computing resources in the brain). Much evidence suggests that the MD system can flexibly store task-relevant information in the short term (e.g., Fedorenko et al., 2013; Freedman et al., 2001; Shashidhara, Mitchell, et al., 2019; Wen et al., 2019; Woolgar et al., 2011). However, evidence from studies on processing mathematics (e.g., Amalric & Dehaene, 2019) and physics (e.g., Cetron et al., 2019; Fischer et al., 2016) further suggests that the MD system can store some domain-specific representations in the long term, perhaps for evolutionarily late-emerging and ontogenetically late-acquired domains of knowledge. Our data add to this body of evidence by showing that the MD system stores and uses information required for code comprehension.

We also show that, instead of being concentrated in one region or a subset of the MD system, code-evoked responses are distributed throughout the MD system. This result seems to violate general metabolic and computational efficiency principles that govern much of the brain's architecture (Chklovskii & Koulakov, 2004; Kanwisher, 2010): if some MD neurons are, at least in part, functionally specialized to process computer code, we would expect them to be located next to each other. Three possibilities are worth considering. First, selectivity for code comprehension in a subset of the MD network may only emerge with years of experience (e.g., in professional programmers). Participants in our experiments were all proficient in the target programming language but most had only a few years of experience with it. Second, code-selective subsets of the MD network may be detectable at higher spatial resolution, using invasive methods like electrocorticography (Parvizi & Kastner, 2018) or single-cell recordings (Mukamel & Fried, 2012). And third, perhaps the need to flexibly solve novel problems throughout one's life prevents the 'crystallization' of specialized subnetworks within the MD cortex.

The language system is functionally conservative

We found that the language system does not respond consistently during code comprehension in spite of numerous similarities between code and natural languages (Fedorenko et al., 2019). Perhaps the most salient similarity between these input types is their syntactic/combinatorial structure. Some accounts of language processing claim that syntactic operations that support language processing are highly abstract and insensitive to the nature of the to-be-combined units (e.g., Berwick et al., 2013; Fitch et al., 2005; Hauser et al., 2002). Such accounts predict that the mechanisms supporting structure processing in language should also get engaged when we process structure in other domains, including computer code. Prior work has already put into question this idea in its broadest form: processing music, whose hierarchical structure has long been noted to have parallels with linguistic syntax (e.g., Lerdahl & Jackendoff, 1996), does not engage the language system (e.g., Fedorenko, McDermott, et al., 2012; Rogalsky et al., 2011). However, music does not have semantics (Slevc, 2012); perhaps some degree of meaningfulness is required to engage linguistic syntactic mechanisms? Our findings, along with prior findings from math and logic (Amalric & Dehaene, 2019; Monti et al., 2009, 2012), argue against this possibility: the language system does not respond to meaningful structured input that is non-linguistic. The lack of the language system engagement during code comprehension adds to the body of work that demonstrates high input selectivity of these regions (Fedorenko et al., 2011; Jouravlev et al., 2019; Monti et al., 2009, 2012; Pritchett et al., 2018).

Although the language system does not appear to support code comprehension, it may play a role in learning to program (Prat et al., 2020). Studies advocating the 'coding as another language' approach (Bers, 2019, 2018; Sullivan & Bers, 2019) have found that treating coding as a meaning-making activity rather than merely a problem-solving skill had a positive impact on both teaching and learning to program in the classroom (Hassenfeld et al., 2020; Hassenfeld & Bers, 2020). It is therefore possible that the language system and/or the general semantic system plays a role in learning to process computer code, especially in children, when the language system is still developing. This idea remains to be empirically evaluated in future studies.

Limitations of scope

The stimuli used in our study were short and only included a few basic elements of control flow (such as *for* loops and *if* statements). Furthermore, we focused on code comprehension, which is a necessary but not sufficient component of many other programming activities, such as code generation, editing, and debugging. Future work should investigate changes in brain activity during the processing and generation of more complex code structures, such as functions, objects, and large multi-component programs. Just like narrative processing recruits systems outside the regions that support single sentence processing (Baldassano et al., 2018; Blank & Fedorenko, 2019; Ferstl et al., 2008; Jacoby & Fedorenko, 2018; Lerner

et al., 2011; Simony et al., 2016), reading more complex pieces of code might recruit an extended, or a different, set of brain regions. Furthermore, as noted above, investigations of expert programmers may reveal changes in how programming knowledge and use are instantiated in the mind and brain as a function of increasing amount of domain-relevant experience.

Overall, we provide evidence that code comprehension consistently recruits the MD system—which subserves cognitive processing across multiple cognitive domains—but does not consistently engage the language system, in spite of numerous similarities between natural and programming languages. By isolating neural activity specific to code comprehension, we pave the way for future studies examining the cognitive and neural correlates of programming and contribute to the broader literature on the neural systems that support novel cognitive tools.

Method

Participants

For Experiment 1, we recruited 25 participants (15 women, mean age = 23.0 years, SD = 3.0). Average age at which participants started to program was 16 years (SD = 2.6); average number of years spent programming was 6.3 (SD=3.8). In addition to Python, 20 people also reported some knowledge of Java, 18 people reported knowledge of C/C++, 4 of functional languages, and 20 of numerical languages like Matlab and R. Twenty-three participants were right-handed, one was ambidextrous, and one was left-handed (as assessed by Oldfield's (1971) handedness questionnaire); the left-handed participant had a right-lateralized language system and was excluded from the analyses, leaving 24 participants (all of whom had left-lateralized language regions). Participants also reported their knowledge of foreign languages and completed a one-hour-long Python proficiency test (available on the paper's website).

For Experiment 2, we recruited 21 participants (13 women, mean age = 22.5 years, SD = 2.8). In addition to ScratchJr, 8 people also reported some knowledge of Python, 6 people reported knowledge of Java, 9 people reported knowledge of C/C++, 1 of functional languages, and 14 of numerical languages like Matlab and R (one participant did not complete the programming questionnaire). Twenty were right-handed and one was ambidextrous; all participants had left-lateralized language regions, as evaluated with the language localizer task (see below). Two participants from Experiment 2 had to be excluded due to excessive motion during the MRI scan, leaving 19 participants.

All participants were recruited from MIT, Tufts University, and the surrounding community and paid for participation. All were native speakers of English, had normal or corrected to normal vision, and reported working knowledge of Python or ScratchJr, respectively. The protocol for the study was approved by MIT's Committee on the Use of Humans as Experimental Subjects (COUHES). All participants gave written informed consent in accordance with protocol requirements.

Design, materials, and procedure

All participants completed the main program comprehension task, a spatial working memory localizer task aimed at identifying the multiple demand (MD) brain regions (Fedorenko et al., 2011) and a language localizer task aimed at identifying language-responsive brain regions (Fedorenko et al., 2010).

The **program comprehension task in Experiment 1** included three conditions: programs in Python with English identifiers, programs in Python with Japanese identifiers, and sentence versions of those programs (visually presented). The full list of problems can be found on the paper's website, https://github.com/ALFA-group/neural-program-comprehension. Each participant saw 72 problems, and any given participant saw only one version of a problem. Half of the problems required performing mathematical operations, and the other half required string manipulations. In addition, both math and string-manipulation problems varied in program structure: 1/3 of the problems of each type included only sequential statements, 1/3 included a *for* loop, and 1/3 included an *if* statement.

During each trial, participants were instructed to read the problem statement and press a button when they were ready to respond (the minimum processing time was restricted to 5 s and the maximum to 50 s; mean reading time was 19 s). Once they pressed the button, four response options were revealed, and participants had to indicate their response by pressing one of four buttons on a button box. The response screen was presented for 5 s (see **Supplementary Figure 9A** for a schematic of trial structure). Each run consisted of 6 trials (2 per condition), and 3 fixation blocks (at the beginning and end of the run, and after the third trial), each lasting 10 s. A run lasted, on average, 176 s (SD = 34 s), and each participant completed 12 runs. Condition order was counterbalanced across runs and participants.

The **program comprehension task in Experiment 2** included two conditions: short programs in ScratchJr and the sentence versions of those programs (visually presented). ScratchJr is a language designed to teach programming concepts to kids (Bers, 2018): users can create events and sequences of events (stories) with a set of characters and actions. The full list of problems used in the study can be found on the paper's website. Each participant saw 24 problems, and any given participant saw only one version of a problem. Furthermore, problems varied in the complexity of the code snippet (3 levels of difficulty; 8 problems at each level).

During each trial, participants were presented with a fixation cross for 4 s, followed by a description (either a code snippet or a sentence) to read for 8 s. The presentation of the description was followed by 5-9 s of fixation, and then by a video (average duration: 4.13 s, SD: 1.70 s) that either did or did not match the description. Participants had to indicate whether the video matched the description by pressing one of two buttons on a button box in the scanner. The response window started with the onset of the video and included a 4 s period after the video offset. A trial lasted, on average, 27.46 s (SD = 2.54 s; see **Supplementary Figure 9B** for a schematic of trial structure). Each run consisted of 6 trials (3 per condition), and a 10 s fixation at the start of the trial as well as at the end of the trial. A run lasted, on average, 184.75 s (SD = 3.86 s); each participant completed 4 runs. Condition order was counterbalanced across runs and participants.

The **spatial working memory task** was conducted in order to identify the MD system within individual participants. Participants had to keep track of four (easy condition) or eight (hard condition) sequentially presented locations in a 3 × 4 grid (**Figure 1B**; Fedorenko et al., 2011). In both conditions, they performed a two-alternative forced-choice task at the end of each trial to indicate the set of locations they just saw. The hard > easy contrast has been previously shown to reliably activate bilateral fronal and parietal MD regions (Assem et al., 2019; Blank et al., 2014; Fedorenko et al., 2013). Numerous studies have shown that the same brain regions are activated by diverse executively-demanding tasks (Duncan & Owen, 2000; Fedorenko et al., 2013; Hugdahl et al., 2015; Shashidhara, Mitchell, et al., 2019; Woolgar et al., 2011). Stimuli were presented in the center of the screen across four steps. Each step lasted 1000 ms and revealed one location on the grid in the easy condition, and two locations in the hard condition. Each stimulus was followed by a choice-selection step, which showed two grids side by side. One grid contained the locations shown across the previous four steps, while the other contained an incorrect set of locations. Participants were asked to press one of two buttons to choose the grid that showed the correct locations. Condition order was counterbalanced across runs. Experimental blocks lasted 32 s (with 4 trials per block), and fixation blocks lasted 16 s. Each run (consisting of 4 fixation blocks and 12 experimental blocks) lasted 448 s. Each participant completed 2 runs.

The **language localizer task** was conducted in order to identify the language system within individual participants. Participants read sentences (e.g., NOBODY COULD HAVE PREDICTED THE EARTHQUAKE IN THIS PART OF THE COUNTRY) and lists of unconnected, pronounceable nonwords (e.g., U BIZBY ACWORRILY MIDARAL MAPE LAS POME U TRINT WEPS WIBRON PUZ) in a blocked design. Each stimulus consisted of twelve words/nonwords. For details of how the language materials were constructed, see Fedorenko et al. (2010). The materials are available at http://web.mit.edu/evelina9/www/funcloc/funcloc_localizers.html. The sentences > nonword-lists contrast has been previously shown to reliably activate left-lateralized fronto-temporal language processing regions and to be robust to changes in the materials, task, and modality of presentation (Fedorenko et al., 2010; Mahowald & Fedorenko, 2016; Scott et al., 2017). Stimuli were presented in the center of the screen, one word/nonword at a time, at the rate of 450 ms per word/nonword. Each stimulus was preceded by a 100 ms blank screen and followed by a 400 ms screen showing a picture of a finger pressing a button, and a blank screen for another 100 ms, for a total trial duration of 6 s. Participants were asked to press a button whenever they saw the picture of a finger pressing a button. This task was included to help participants stay alert. Condition order was counterbalanced across runs. Experimental blocks lasted 18 s (with 3 trials per block), and fixation blocks lasted 14 s. Each run (consisting of 5 fixation blocks and 16 experimental blocks) lasted 358 s. Each participant completed 2 runs.

fMRI data acquisition

Structural and functional data were collected on the whole-body, 3 Tesla, Siemens Trio scanner with a 32-channel head coil, at the Athinoula A. Martinos Imaging Center at the McGovern Institute for Brain Research at MIT. T1-weighted structural images were collected in 176 sagittal slices with 1mm isotropic voxels (TR = 2,530 ms, TE = 3.48 ms). Functional, blood oxygenation level dependent (BOLD), data were acquired using an EPI sequence (with a 90o flip angle and using GRAPPA with an acceleration factor of 2), with the following acquisition parameters: thirty-one 4mm thick near-axial slices acquired in the interleaved order (with 10% distance factor), 2.1 mm × 2.1 mm in-plane resolution, FoV in the phase encoding (A>>P) direction 200mm and matrix size 96 mm × 96 mm, TR = 2,000 ms and TE = 30 ms. The first 10 s of each run were excluded to allow for steady state magnetization.

fMRI data preprocessing

MRI data were analyzed using SPM12 and custom MATLAB scripts (available in the form of an SPM toolbox from http://www.nitrc.org/projects/spm_ss). Each participant's data were motion corrected and then normalized into a common brain space (the Montreal Neurological Institute (MNI) template) and resampled into 2mm isotropic voxels. The data were then smoothed with a 4mm FWHM Gaussian filter and high-pass filtered (at 128 s). Effects were estimated using a General Linear Model (GLM) in which each experimental condition was modeled with a boxcar function convolved with the canonical hemodynamic response function (HRF). For the localizer experiments, we modeled the entire blocks. For the Python program comprehension experiment, we modeled the period from the onset of the code/sentence problem and until the button press (the responses were modeled as a separate condition; see **Supplementary Figure 9A**); for the ScratchJr program comprehension experiment, we modeled the period of the code/sentence presentation (the video and the response were modeled as a separate condition; see **Supplementary Figure 9B**).

Defining MD and language functional regions of interest (fROIs)

The fROI analyses examined responses in individually defined MD and language fROIs (functional regions of interest). These fROIs were defined using the Group-constrained Subject-Specific (GSS) approach (Fedorenko et al. 2010; Julian et al. 2012) where a set of spatial masks, or parcels, is combined with each individual subject's localizer activation map, to constrain the definition of individual fROIs. The parcels delineate the expected gross locations of activations for a given contrast based on prior work and large numbers of participants and are sufficiently large to encompass the variability in the locations of individual activations. For the MD system, we used a set of 20 parcels (10 in each hemisphere) derived from a group-level probabilistic activation overlap map for the hard > easy spatial working memory contrast in 197 participants. The parcels included regions in frontal and parietal lobes, as well as a region in the anterior cingulate cortex. For the language system, we used a set of six parcels derived from a group-level probabilistic activation overlap map for the sentences > nonwords contrast in 220 participants. The parcels included two regions in the left inferior frontal gyrus (LIFG, LIFGorb), one in the left middle frontal gyrus (LMFG), two in the left temporal lobe (LAntTemp and LPostTemp), and one extending into the angular gyrus (LAngG). Both sets of parcels are available on the paper's website; see **Supplementary Figure 2** for labeled images of MD and language parcels. Within each parcel, we selected the top 10% most localizer-responsive voxels, based on the *t* values (see, e.g., Figure 1 in Blank et al. (2014), for sample MD and language fROIs). Individual fROIs defined this way were then used for subsequent analyses that examined responses to code comprehension.

Examining the functional response profiles of the MD and language fROIs

Univariate analyses. We evaluated MD and language system responses by estimating their response magnitudes to the conditions of interest using individually defined fROIs (see above). For each fROI in each participant, we averaged the responses across voxels to get a single value for each of the conditions (the responses to the localizer conditions were estimated using an across-runs cross-validation procedure, where one run was used to define the fROI and the other to estimate the response magnitudes, then the procedure was repeated switching which run was used for fROI definition vs. response estimation, and finally the estimates were averaged to derive a single value per condition per fROI per participant). We then ran a linear mixed-effect regression model to compare the responses to the critical code problem condition with (a) the responses to the sentence problem condition from the critical task, and (b) the responses to the nonword reading condition from the language localizer task. We included *condition* as a fixed effect and

participant and *fROI* as random effects. For the MD system, we additionally tested the main (fixed) effect of *hemisphere* and the interaction between *hemisphere* and c*ondition*. For follow-up analyses, we used the variable of interest (*problem type/structure/identifier language*) as a fixed effect and *participant* and *fROI* as random effects. For fROI analyses, we used *condition* as a fixed effect and *participant* as a random effect. The analyses were run using the *lmer* function from the *lme4* R package (Bates et al., 2015); statistical significance of the effects was evaluated using the *lmerTest* package (Kuznetsova et al., 2017).

Spatial correlation analyses. To further examine the similarity of the fine-grained patterns of activation between conditions in the language system, we calculated voxel-wise spatial correlations in activation magnitudes within the code problem condition (between odd and even runs), within the sentence problem condition (between odd and even runs), between these two conditions (we used odd and even run splits here, too, to match the amount of data for the within- vs. between-condition comparisons, and averaged the correlation values across the different splits), and between these two critical conditions and each of the sentence and nonword reading conditions from the language localizer. The correlation values were calculated for voxels in each participant's language fROIs, and then averaged across participants and fROIs for plotting. We also used the *lme4* R package to calculate statistical differences between spatial correlation values (with Participant and fROI as random effects); for this analysis, the correlation values were Fischer-transformed.

Whole-brain analyses

For each of the critical experiments (Python and ScratchJr), we conducted (a) the Group-constrained Subject-Specific (GSS) analysis (Fedorenko et al., 2010; Julian et al., 2012), and (b) the traditional random effects group analysis (Holmes & Friston, 1998) using the code problems > sentence problems contrast. The analyses were performed using the spm_ss toolbox (http://www.nitrc.org/projects/spm_ss), which interfaces with SPM and the CONN toolbox (https://www.nitrc.org/projects/conn).

Acknowledgements

We would like to acknowledge the Athinoula A. Martinos Imaging Center at the McGovern Institute for Brain Research at MIT and its support team (Steve Shannon, Atsushi Takahashi, and Dima Ayyash), Rachel Ryskin for advice on statistics, Alfonso Nieto-Castañón for help with analyses, ALFA group at CSAIL for helpful discussions on the experiment design, and Josef Affourtit, Yev Diachek, and Matt Siegelman (EvLab), and Ruthi Aladjem, Claudia Mihm, and Kaitlyn Leid (DevTech research group at Tufts University) for technical support during experiment design and data collection. This research was supported by NSF award #1744809 to M.B. and E.F. E.F. was additionally supported by funds from the Department of the Brain and Cognitive Sciences at MIT and the McGovern Institute for Brain Research.

Data/Code Availability

Materials used for the critical programming tasks, fROI responses in individual participants (used for generating the main results), and analysis code files are available on the paper's website https://github.com/ALFA-group/neural-program-comprehension. Raw/preprocessed fMRI files are available upon request.

References

Amalric, M., & Dehaene, S. (2019). A distinct cortical network for mathematical knowledge in the human brain. *NeuroImage*, *189*, 19–31. https://doi.org/10.1016/j.neuroimage.2019.01.001

Anderson, A. J., Lalor, E. C., Lin, F., Binder, J. R., Fernandino, L., Humphries, C. J., Conant, L. L., Raizada, R. D. S., Grimm, S., & Wang, X. (2019). Multiple Regions of a Cortical Network Commonly Encode the Meaning of Words in Multiple Grammatical Positions of Read Sentences. *Cerebral Cortex (New York, N.Y.: 1991)*, *29*(6), 2396–2411. https://doi.org/10.1093/cercor/bhy110

Assem, M., Glasser, M. F., Essen, D. C. V., & Duncan, J. (2019). A Domain-general Cognitive Core defined in Multimodally Parcellated Human Cortex. *BioRxiv*, 517599. https://doi.org/10.1101/517599

Baldassano, C., Hasson, U., & Norman, K. A. (2018). Representation of Real-World Event Schemas during Narrative Perception. *Journal of Neuroscience*, *38*(45), 9689–9699. https://doi.org/10.1523/JNEUROSCI.0251-18.2018

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bautista, A., & Wilson, S. M. (2016). Neural responses to grammatically and lexically degraded speech. *Language, Cognition and Neuroscience*, *31*(4), 567–574. https://doi.org/10.1080/23273798.2015.1123281

Bemis, D. K., & Pylkkänen, L. (2013). Basic Linguistic Composition Recruits the Left Anterior Temporal Lobe and Left Angular Gyrus During Both Listening and Reading. *Cerebral Cortex*, *23*(8), 1859–1873. https://doi.org/10.1093/cercor/bhs170

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. https://doi.org/10.1027/1614-0001/a000052

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3

Bers, M. U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2094–2102. https://doi.org/10.1109/EDUCON.2018.8363498

Bers, M. U., & Resnick, M. (2015). *The Official ScratchJr Book: Help Your Kids Learn to Code* (1 edition). No Starch Press.

Berwick, R. C., Friederici, A. D., Chomsky, N., & Bolhuis, J. J. (2013). Evolution, brain, and the nature of language. *Trends in Cognitive Sciences*, *17*(2), 89–98. https://doi.org/10.1016/j.tics.2012.12.002

Binder, J. R., Desai, R. H., Graves, W. W., & Conant, L. L. (2009). Where is the semantic system? A critical review and meta-analysis of 120 functional neuroimaging studies. *Cerebral Cortex (New York, N.Y.: 1991)*, *19*(12), 2767–2796. https://doi.org/10.1093/cercor/bhp055

Blank, I. A., & Fedorenko, E. (2019). No evidence for functional distinctions across fronto-temporal language regions in their temporal receptive windows. *BioRxiv*, 712372. https://doi.org/10.1101/712372

Blank, I. A., Kanwisher, N., & Fedorenko, E. (2014). A functional dissociation between language and multiple-demand systems revealed in patterns of BOLD signal fluctuations. *Journal of Neurophysiology*, *112*(5), 1105–1118. https://doi.org/10.1152/jn.00884.2013

Blank, I. A., Kiran, S., & Fedorenko, E. (2017). Can neuroimaging help aphasia researchers? Addressing generalizability, variability, and interpretability. *Cognitive Neuropsychology*, *34*(6), 377–393. https://doi.org/10.1080/02643294.2017.1402756

Braga, R. M., Van Dijk, K. R. A., Polimeni, J. R., Eldaief, M. C., & Buckner, R. L. (2019). Parallel distributed networks resolved at high resolution reveal close juxtaposition of distinct regions. *Journal of Neurophysiology*, *121*(4), 1513–1534. https://doi.org/10.1152/jn.00808.2018

Brett, M., Johnsrude, I. S., & Owen, A. M. (2002). The problem of functional localization in the human brain. *Nature Reviews. Neuroscience*, *3*(3), 243–249. https://doi.org/10.1038/nrn756

Castelhano, J., Duarte, I. C., Ferreira, C., Duraes, J., Madeira, H., & Castelo-Branco, M. (2019). The role of the insula in intuitive expert bug detection in computer code: An fMRI study. *Brain Imaging and Behavior*, *13*(3), 623–637. https://doi.org/10.1007/s11682-018-9885-1

Cetron, J. S., Connolly, A. C., Diamond, S. G., May, V. V., Haxby, J. V., & Kraemer, D. J. M. (2019). Decoding individual differences in STEM learning from functional MRI data. *Nature Communications*, *10*(1), 1–10. https://doi.org/10.1038/s41467-019-10053-y

Chklovskii, D. B., & Koulakov, A. A. (2004). Maps in the brain: What can we learn from them? *Annual Review of Neuroscience*, *27*, 369–392. https://doi.org/10.1146/annurev.neuro.27.070203.144226

Corballis, P. M. (2003). Visuospatial processing and the right-hemisphere interpreter. *Brain and Cognition*, *53*(2), 171–176. https://doi.org/10.1016/s0278-2626(03)00103-9

Dalbey, J., & Linn, M. (1985). The Demands and Requirements of Computer Programming: A Literature Review. *Journal of Educational Computing Research*, *1*. https://doi.org/10.2190/BC76-8479-YM0X-7FUA

Deniz, F., Nunez-Elizalde, A. O., Huth, A. G., & Gallant, J. L. (2019). The Representation of Semantic Information Across Human Cerebral Cortex During Listening Versus Reading Is Invariant to Stimulus Modality. *Journal of Neuroscience*, *39*(39), 7722–7736. https://doi.org/10.1523/JNEUROSCI.0675-19.2019

Duncan, J. (2010). The multiple-demand (MD) system of the primate brain: Mental programs for intelligent behaviour. *Trends in Cognitive Sciences*, *14*(4), 172–179. https://doi.org/10.1016/j.tics.2010.01.004

Duncan, J. (2013). The Structure of Cognition: Attentional Episodes in Mind and Brain. *Neuron*, *80*(1), 35–50. https://doi.org/10.1016/j.neuron.2013.09.015

Duncan, J., & Owen, A. M. (2000). Common regions of the human frontal lobe recruited by diverse cognitive demands. *Trends in Neurosciences*, *23*(10), 475–483. https://doi.org/10.1016/S0166-2236(00)01633-7

Ensmenger, N. (2012). The Digital Construction of Technology: Rethinking the History of Computers in Society. *Technology and Culture*, *53*(4), 753–776. https://doi.org/10.1353/tech.2012.0126

Fakhoury, S., Ma, Y., Arnaoudova, V., & Adesope, O. (2018). The Effect of Poor Source Code Lexicon and Readability on Developers' Cognitive Load. *Proceedings of the 26th Conference on Program Comprehension*, 286–296. https://doi.org/10.1145/3196321.3196347

Fedorenko, E., Behr, M. K., & Kanwisher, N. (2011). Functional specificity for high-level linguistic processing in the human brain. *Proceedings of the National Academy of Sciences*, *108*(39), 16428–16433. https://doi.org/10.1073/pnas.1112937108

Fedorenko, E., & Blank, I. A. (2020). Broca's Area Is Not a Natural Kind. *Trends in Cognitive Sciences*. https://doi.org/10.1016/j.tics.2020.01.001

Fedorenko, E., Blank, I., Siegelman, M., & Mineroff, Z. (2020). Lack of selectivity for syntax relative to word meanings throughout the language network. *BioRxiv*, 477851. https://doi.org/10.1101/477851

Fedorenko, E., Duncan, J., & Kanwisher, N. (2013). Broad domain generality in focal regions of frontal and parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(41), 16616–16621. https://doi.org/10.1073/pnas.1315235110

Fedorenko, E., Hsieh, P.-J., Nieto-Castañón, A., Whitfield-Gabrieli, S., & Kanwisher, N. (2010). New method for fMRI investigations of language: Defining ROIs functionally in individual subjects. *Journal of Neurophysiology*, *104*(2), 1177–1194. https://doi.org/10.1152/jn.00032.2010

Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The Language of Programming: A Cognitive Perspective. *Trends in Cognitive Sciences*, *23*(7), 525–528. https://doi.org/10.1016/j.tics.2019.04.010

Fedorenko, E., & Kanwisher, N. (2009). Neuroimaging of Language: Why Hasn't a Clearer Picture Emerged? *Language and Linguistics Compass*, *3*(4), 839–865. https://doi.org/10.1111/j.1749-818X.2009.00143.x

Fedorenko, E., McDermott, J. H., Norman-Haignere, S., & Kanwisher, N. (2012). Sensitivity to musical structure in the human brain. *Journal of Neurophysiology*, *108*(12), 3289–3300. https://doi.org/10.1152/jn.00209.2012

Fedorenko, E., Nieto-Castañon, A., & Kanwisher, N. (2012). Lexical and syntactic representations in the brain: An fMRI investigation with multi-voxel pattern analyses. *Neuropsychologia*, *50*(4), 499–513. https://doi.org/10.1016/j.neuropsychologia.2011.09.014

Fedorenko, E., & Thompson-Schill, S. L. (2014). Reworking the language network. *Trends in Cognitive Sciences*, *18*(3), 120–126. https://doi.org/10.1016/j.tics.2013.12.006

Ferstl, E. C., Neumann, J., Bogler, C., & von Cramon, D. Y. (2008). The extended language network: A meta-analysis of neuroimaging studies on text comprehension. *Human Brain Mapping*, *29*(5), 581–593. https://doi.org/10.1002/hbm.20422

Fischer, J., Mikhael, J. G., Tenenbaum, J. B., & Kanwisher, N. (2016). Functional neuroanatomy of intuitive physical inference. *Proceedings of the National Academy of Sciences*, *113*(34), E5072–E5081. https://doi.org/10.1073/pnas.1610344113

Fitch, W. T., Hauser, M. D., & Chomsky, N. (2005). The evolution of the language faculty: Clarifications and implications. *Cognition*, *97*(2), 179–210. https://doi.org/10.1016/j.cognition.2005.02.005

Floyd, B., Santander, T., & Weimer, W. (2017). Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 175–186. https://doi.org/10.1109/ICSE.2017.24

Freedman, D. J., Riesenhuber, M., Poggio, T., & Miller, E. K. (2001). Categorical representation of visual stimuli in the primate prefrontal cortex. *Science (New York, N.Y.)*, *291*(5502), 312–316. https://doi.org/10.1126/science.291.5502.312

Goel, V. (2007). Anatomy of deductive reasoning. *Trends in Cognitive Sciences*, *11*(10), 435–441. https://doi.org/10.1016/j.tics.2007.09.003

Goel, V., & Dolan, R. J. (2001). Functional neuroanatomy of three-term relational reasoning. *Neuropsychologia*, *39*(9), 901–909. https://doi.org/10.1016/S0028-3932(01)00024-0

Guzdial, M. (2008). EducationPaving the way for computational thinking. *Communications of the ACM*, *51*(8), 25–27. https://doi.org/10.1145/1378704.1378713

Hassenfeld, Z. R., & Bers, M. U. (2020). Debugging the Writing Process: Lessons From a Comparison of Students' Coding and Writing Practices. *International Literacy Association*, 1–12. https://doi.org/10.1002/trtr.1885

Hassenfeld, Z. R., Govind, M., Ruiter, L. E. D., & Bers, M. U. (2020). If You Can Program, You Can Write: Learning Introductory Programming Across Literacy Levels. *Journal of Information Technology Education: Research*, *19*, 065–085.

Hasson, U., Chen, J., & Honey, C. J. (2015). Hierarchical process memory: Memory as an integral component of information processing. *Trends in Cognitive Sciences*, *19*(6), 304–313. https://doi.org/10.1016/j.tics.2015.04.006

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002). The Faculty of Language: What Is It, Who Has It, and How Did It Evolve? *Science*, *298*(5598), 1569–1579. https://doi.org/10.1126/science.298.5598.1569

Hermans, F., & Aldewereld, M. (2017). Programming is Writing is Programming. *Companion to the First International Conference on the Art, Science and Engineering of Programming*, 33:1–33:8. https://doi.org/10.1145/3079368.3079413

Holmes, A. P., & Friston, K. J. (1998). Generalisability, Random Effects & Population Inference. *NeuroImage*. https://doi.org/10.1016/S1053-8119(18)31587-8

Huang, Y., Liu, X., Krueger, R., Santander, T., Hu, X., Leach, K., & Weimer, W. (2019). Distilling Neural Representations of Data Structure Manipulation using fMRI and fNIRS. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 396–407. https://doi.org/10.1109/ICSE.2019.00053

Hugdahl, K. (2011). Hemispheric asymmetry: Contributions from brain imaging. *Wiley Interdisciplinary Reviews. Cognitive Science*, *2*(5), 461–478. https://doi.org/10.1002/wcs.122

Hugdahl, K., Raichle, M. E., Mitra, A., & Specht, K. (2015). On the existence of a generalized non-specific task-dependent network. *Frontiers in Human Neuroscience*, *9*. https://doi.org/10.3389/fnhum.2015.00430

Ikutani, Y., Kubo, T., Nishida, S., Hata, H., Matsumoto, K., Ikeda, K., & Nishimoto, S. (2020). Expert programmers have fine-tuned cortical representations of source code. *BioRxiv*, 2020.01.28.923953. https://doi.org/10.1101/2020.01.28.923953

Ikutani, Y., & Uwano, H. (2014). Brain activity measurement during program comprehension with NIRS. *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 1–6. https://doi.org/10.1109/SNPD.2014.6888727

Jacoby, N., & Fedorenko, E. (2018). Discourse-level comprehension engages medial frontal Theory of Mind brain regions even for expository texts. *Language, Cognition and Neuroscience*, *0*(0), 1–17. https://doi.org/10.1080/23273798.2018.1525494

Jouravlev, O., Zheng, D., Balewski, Z., Pongos, A. L. A., Levan, Z., Goldin-Meadow, S., & Fedorenko, E. (2019). Speech-accompanying gestures are not processed by the language-processing mechanisms. *Neuropsychologia*, *132*, 107132. https://doi.org/10.1016/j.neuropsychologia.2019.107132

Julian, J. B., Fedorenko, E., Webster, J., & Kanwisher, N. (2012). An algorithmic method for functionally defining regions of interest in the ventral visual pathway. *NeuroImage*, *60*(4), 2357–2364. https://doi.org/10.1016/j.neuroimage.2012.02.055

Kanwisher, N. (2010). Functional specificity in the human brain: A window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences*, *107*(25), 11163–11170. https://doi.org/10.1073/pnas.1005062107

Keller, T. A., Carpenter, P. A., & Just, M. A. (2001). The Neural Bases of Sentence Comprehension: A fMRI Examination of Syntactic and Lexical Processing. *Cerebral Cortex*, *11*(3), 223–237. https://doi.org/10.1093/cercor/11.3.223

Kroll, J. F., Dussias, P. E., Bice, K., & Perrotti, L. (2015). Bilingualism, Mind, and Brain. *Annual Review of Linguistics*, *1*, 377–394. https://doi.org/10.1146/annurev-linguist-030514-124937

Krueger, R., Huang, Y., Liu, X., Santander, T., Weimer, W., & Leach, K. (2020). Neurological Divide: An fMRI Study of Prose and Code Writing. *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, 13.

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software*, *82*(1), 1–26. https://doi.org/10.18637/jss.v082.i13

Lerdahl, F., & Jackendoff, R. S. (1996). *A Generative Theory of Tonal Music*. MIT Press.

Lerner, Y., Honey, C. J., Silbert, L. J., & Hasson, U. (2011). Topographic Mapping of a Hierarchy of Temporal Receptive Windows Using a Narrated Story. *The Journal of Neuroscience*, *31*(8), 2906–2915. https://doi.org/10.1523/JNEUROSCI.3684-10.2011

Mahowald, K., & Fedorenko, E. (2016). Reliable individual-level neural markers of high-level language processing: A necessary precursor for relating neural variability to behavioral and genetic variability. *NeuroImage*, *139*, 74–93. https://doi.org/10.1016/j.neuroimage.2016.05.073

Mather, M., Cacioppo, J. T., & Kanwisher, N. (2013). How fMRI Can Inform Cognitive Theories. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, *8*(1), 108–113. https://doi.org/10.1177/1745691612469037

McNamara, W. J. (1967). The Selection of Computer Personnel: Past, Present, Future. *Proceedings of the Fifth SIGCPR Conference on Computer Personnel Research*, 52–56. https://doi.org/10.1145/1142662.1142667

Michalka, S. W., Kong, L., Rosen, M. L., Shinn-Cunningham, B. G., & Somers, D. C. (2015). Short-Term Memory for Space and Time Flexibly Recruit Complementary Sensory-Biased Frontal Lobe Attention Networks. *Neuron*, *87*(4), 882–892. https://doi.org/10.1016/j.neuron.2015.07.028

Micheloyannis, S., Sakkalis, V., Vourkas, M., Stam, C. J., & Simos, P. G. (2005). Neural networks involved in mathematical thinking: Evidence from linear and non-linear analysis of electroencephalographic activity. *Neuroscience Letters*, *373*(3), 212–217. https://doi.org/10.1016/j.neulet.2004.10.005

Miller, E. K., & Cohen, J. D. (2001). An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience*, *24*(1), 167–202. https://doi.org/10.1146/annurev.neuro.24.1.167

Miller, E. K., Erickson, C. A., & Desimone, R. (1996). Neural Mechanisms of Visual Working Memory in Prefrontal Cortex of the Macaque. *Journal of Neuroscience*, *16*(16), 5154–5167. https://doi.org/10.1523/JNEUROSCI.16-16-05154.1996

Mineroff, Z., Blank, I. A., Mahowald, K., & Fedorenko, E. (2018). A robust dissociation among the language, multiple demand, and default mode networks: Evidence from inter-region correlations in effect size. *Neuropsychologia*, *119*, 501–511. https://doi.org/10.1016/j.neuropsychologia.2018.09.011

Mitchell, D. J., Bell, A. H., Buckley, M. J., Mitchell, A. S., Sallet, J., & Duncan, J. (2016). A Putative Multiple-Demand System in the Macaque Brain. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *36*(33), 8574–8585. https://doi.org/10.1523/JNEUROSCI.0810-16.2016

Mollica, F., Siegelman, M., Diachek, E., Piantadosi, S. T., Mineroff, Z., Futrell, R., Kean, H., Qian, P., & Fedorenko, E. (2020). Composition is the Core Driver of the Language-selective Network. *Neurobiology of Language*, 1–30. https://doi.org/10.1162/nol_a_00005

Monti, M. M., Osherson, D. N., Martinez, M. J., & Parsons, L. M. (2007). Functional neuroanatomy of deductive inference: A language-independent distributed network. *NeuroImage*, *37*(3), 1005–1016. https://doi.org/10.1016/j.neuroimage.2007.04.069

Monti, M. M., Parsons, L. M., & Osherson, D. N. (2009). The boundaries of language and thought in deductive inference. *Proceedings of the National Academy of Sciences*, *106*(30), 12554–12559. https://doi.org/10.1073/pnas.0902422106

Monti, M. M., Parsons, L. M., & Osherson, D. N. (2012). Thought beyond language: Neural dissociation of algebra and natural language. *Psychological Science*, *23*(8), 914–922. https://doi.org/10.1177/0956797612437427

Mukamel, R., & Fried, I. (2012). Human Intracranial Recordings and Cognitive Neuroscience. *Annual Review of Psychology*, *63*(1), 511–537. https://doi.org/10.1146/annurev-psych-120709-145401

Murnane, J. S. (1993). The psychology of computer languages for introductory programming courses. *New Ideas in Psychology*, *11*(2), 213–228. https://doi.org/10.1016/0732-118X(93)90035-C

Nakagawa, T., Kamei, Y., Uwano, H., Monden, A., Matsumoto, K., & German, D. M. (2014). Quantifying Programmers' Mental Workload During Program Comprehension Based on Cerebral Blood Flow Measurement: A Controlled Experiment. *Companion Proceedings of the 36th International Conference on Software Engineering*, 448–451. https://doi.org/10.1145/2591062.2591098

Nakamura, M., Monden, A., Itoh, T., Matsumoto, K., Kanzaki, Y., & Satoh, H. (2003). Queue-based cost evaluation of mental simulation process in program comprehension. *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717)*, 351–360. https://doi.org/10.1109/METRIC.2003.1232480

Nieto-Castañón, A., & Fedorenko, E. (2012). Subject-specific functional localizers increase sensitivity and functional resolution of multi-subject analyses. *NeuroImage*, *63*(3), 1646–1669. https://doi.org/10.1016/j.neuroimage.2012.06.065

Ormerod, T. (1990). *Human Cognition and Programming*. https://doi.org/10.1016/B978-0-12-350772-3.50009-4

Papert, S. A. (1972). Teaching Children to be Mathematicians Versus Teaching About Mathematics. *International Journal of Mathematical Education in Science and Technology*, *3*(3), 249–262. https://doi.org/10.1080/0020739700030306

Papert, S. A. (1993). *Mindstorms: Children, Computers, And Powerful Ideas* (2 edition). Basic Books.

Parvizi, J., & Kastner, S. (2018). Promises and limitations of human intracranial electroencephalography. *Nature Neuroscience*, *21*(4), 474–483. https://doi.org/10.1038/s41593-018-0108-2

Pea, R. D., & Kurland, D. M. (1984). *On the cognitive effects of learning computer programming*. https://doi.org/10.1016/0732-118X(84)90018-7

Pennington, N., & Grabowski, B. (1990). The tasks of programming. *Psychology of Programming*, 45–62.

Pereira, F., Lou, B., Pritchett, B., Ritter, S., Gershman, S. J., Kanwisher, N., Botvinick, M., & Fedorenko, E. (2018). Toward a universal decoder of linguistic meaning from brain activation. *Nature Communications*, *9*(1), 1–13. https://doi.org/10.1038/s41467-018-03068-4

Perkins, D. N., & Simmons, R. (1988). Patterns of Misunderstanding: An Integrative Model for Science, Math, and Programming. *Review of Educational Research*, *58*(3), 303–326. https://doi.org/10.3102/00346543058003303

Pinel, P., & Dehaene, S. (2009). Beyond Hemispheric Dominance: Brain Regions Underlying the Joint Lateralization of Language and Arithmetic to the Left Hemisphere. *Journal of Cognitive Neuroscience*, *22*(1), 48–66. https://doi.org/10.1162/jocn.2009.21184

Poldrack, R. A. (2006). Can cognitive processes be inferred from neuroimaging data? *Trends in Cognitive Sciences*, *10*(2), 59–63. https://doi.org/10.1016/j.tics.2005.12.004

Poldrack, R. A. (2011). Inferring mental states from neuroimaging data: From reverse inference to large-scale decoding. *Neuron*, *72*(5), 692–697. https://doi.org/10.1016/j.neuron.2011.11.001

Prabhakaran, V., Smith, J. A. L., Desmond, J. E., Glover, G. H., & Gabrieli, J. D. E. (1997). Neural Substrates of Fluid Reasoning: An fMRI Study of Neocortical Activation during Performance of the Raven's Progressive Matrices Test. *Cognitive Psychology*, *33*(1), 43–63. https://doi.org/10.1006/cogp.1997.0659

Prat, C. S., Madhyastha, T. M., Mottarella, M. J., & Kuo, C.-H. (2020). Relating Natural Language Aptitude to Individual Differences in Learning Programming Languages. *Scientific Reports*, *10*(1), 1–10. https://doi.org/10.1038/s41598-020-60661-8

Pritchett, B. L., Hoeflin, C., Koldewyn, K., Dechter, E., & Fedorenko, E. (2018). High-level language processing regions are not engaged in action observation or imitation. *Journal of Neurophysiology*, *120*(5), 2555–2570. https://doi.org/10.1152/jn.00222.2018

Reverberi, C., Shallice, T., D'Agostini, S., Skrap, M., & Bonatti, L. L. (2009). Cortical bases of elementary deductive reasoning: Inference, memory, and metadeduction. *Neuropsychologia*, *47*(4), 1107–1116. https://doi.org/10.1016/j.neuropsychologia.2009.01.004

Rogalsky, C., Rong, F., Saberi, K., & Hickok, G. (2011). Functional anatomy of language and music perception: Temporal and structural factors investigated using functional magnetic resonance imaging. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *31*(10), 3843–3852. https://doi.org/10.1523/JNEUROSCI.4515-10.2011

Sanner, M. F. (1999). Python: A programming language for software integration and development. *Journal of Molecular Graphics & Modelling*, *17*(1), 57–61.

Saxe, R., Brett, M., & Kanwisher, N. (2006). Divide and conquer: A defense of functional localizers. *NeuroImage*, *30*(4), 1088–1096; discussion 1097-1099. https://doi.org/10.1016/j.neuroimage.2005.12.062

Scott, T. L., Gallée, J., & Fedorenko, E. (2017). A new fun and robust version of an fMRI localizer for the frontotemporal language system. *Cognitive Neuroscience*, *8*(3), 167–176. https://doi.org/10.1080/17588928.2016.1201466

Shashidhara, S., Mitchell, D. J., Erez, Y., & Duncan, J. (2019). Progressive Recruitment of the Frontoparietal Multiple-demand System with Increased Task Complexity, Time Pressure, and Reward. *Journal of Cognitive Neuroscience*, *31*(11), 1617–1630. https://doi.org/10.1162/jocn_a_01440

Shashidhara, S., Spronkers, F. S., & Erez, Y. (2019). Individual-subject functional localization increases univariate activation but not multivariate pattern discriminability in the 'multiple-demand' frontoparietal network. *BioRxiv*, 661934. https://doi.org/10.1101/661934

Sheremata, S. L., Bettencourt, K. C., & Somers, D. C. (2010). Hemispheric Asymmetry in Visuotopic Posterior Parietal Cortex Emerges with Visual Short-Term Memory Load. *Journal of Neuroscience*, *30*(38), 12581–12588. https://doi.org/10.1523/JNEUROSCI.2689-10.2010

Siegmund, J., Kästner, C., Apel, S., Parnin, C., Bethmann, A., Leich, T., Saake, G., & Brechmann, A. (2014). Understanding Understanding Source Code with Functional Magnetic Resonance Imaging. *Proceedings of*

the 36th International Conference on Software Engineering, 378–389. https://doi.org/10.1145/2568225.2568252

Siegmund, J., Peitek, N., Parnin, C., Apel, S., Hofmeister, J., Kästner, C., Begel, A., Bethmann, A., & Brechmann, A. (2017). Measuring Neural Efficiency of Program Comprehension. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 140–150. https://doi.org/10.1145/3106237.3106268

Simony, E., Honey, C. J., Chen, J., Lositsky, O., Yeshurun, Y., Wiesel, A., & Hasson, U. (2016). Dynamic reconfiguration of the default mode network during narrative comprehension. *Nature Communications*, *7*(1), 1–13. https://doi.org/10.1038/ncomms12141

Slevc, L. R. (2012). Language and music: Sound, structure, and meaning. *WIREs Cognitive Science*, *3*(4), 483–492. https://doi.org/10.1002/wcs.1186

Sullivan, A., & Bers, M. U. (2019). Computer Science Education in Early Childhood: The Case of ScratchJr. *Journal of Information Technology Education: Innovations in Practice*, *18*, 113–138.

Wen, T., Duncan, J., & Mitchell, D. J. (2019). Representation of task episodes in human cortical networks. *BioRxiv*, 582858. https://doi.org/10.1101/582858

Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, *6*.

Woolgar, A., Duncan, J., Manes, F., & Fedorenko, E. (2018). *Fluid intelligence is supported by the multiple-demand system not the language system*. https://doi.org/10.17863/CAM.22222

Woolgar, A., Thompson, R., Bor, D., & Duncan, J. (2011). Multi-voxel coding of stimuli, rules, and responses in human frontoparietal cortex. *NeuroImage*, *56*(2), 744–752. https://doi.org/10.1016/j.neuroimage.2010.04.035

Supplementary Information

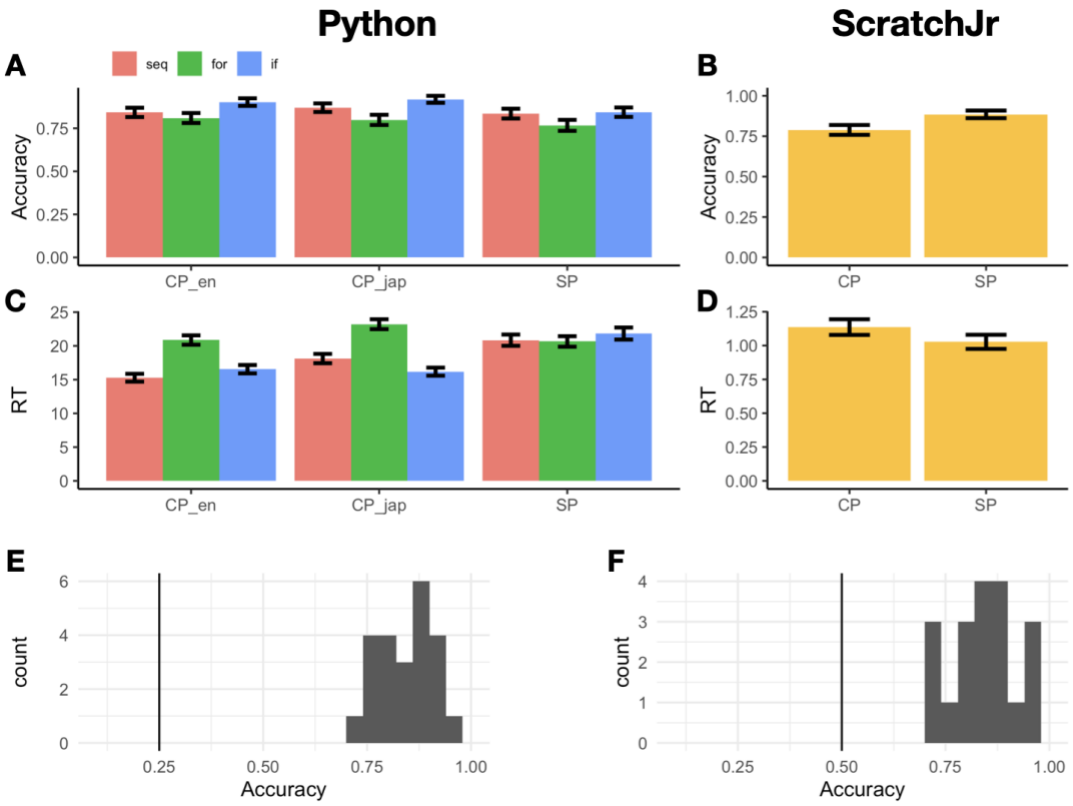

Figure S1. *Behavioral results. (**A**) Python code problems had mean accuracies of 85.1% and 86.2% for the English-identifier (CP_en) and Japanese-identifier (CP_jap) conditions, respectively, and sentence problems (SP) had a mean accuracy of 81.5%. There was no main effect of condition (CP_en, CP_jap, SP), problem structure (seq – sequential, for – for loops, if – if statements), or problem content (math vs. string); however, there was a three-way interaction among Condition (sentence problems > code with English identifiers), Problem Type (string > math), and Problem Structure (for loop > sequential; $p = 0.02$). Accuracy data from one participant had to be excluded due to a bug in the script. (**B**) ScratchJr code problems had a mean accuracy of 78.9%, and sentence problems had a mean accuracy of 88.5% (the difference was significant: $p = 0.007$). (**C**) Python problems with English identifiers had a mean response time (RT) of 17.56 s (SD = 9.05), Python problems with Japanese identifiers had a mean RT of 19.39 s (SD = 10.1), and sentence problems had a mean RT of 21.32 s (SD = 11.6). Problems with Japanese identifiers took longer to answer than problems with English identifiers ($\beta = 3.10$, $p = 0.002$), and so did sentence problems ($\beta = 6.12$, $p < 0.001$). There was also an interaction between Condition (sentence problems > code with English identifiers) and Program Structure (for > seq; $\beta = -5.25$, $p < 0.001$), as well as between Condition (CP_jap > CP_en) and Program Structure (if > seq; $\beta = -2.83$, $p = 0.04$). There was no significant difference in response times between math and string manipulation problems. (**D**) ScratchJr code problems had a mean RT of 1.14 s (SD = 0.86), and sentence problems had a mean RT of 1.03 s (SD = 0.78); the difference was not significant. ScratchJr RTs are reported with respect to video offset. (**E**) Mean accuracies for all Python participants were above chance. (**F**) Mean accuracies for all ScratchJr participants were above chance.*

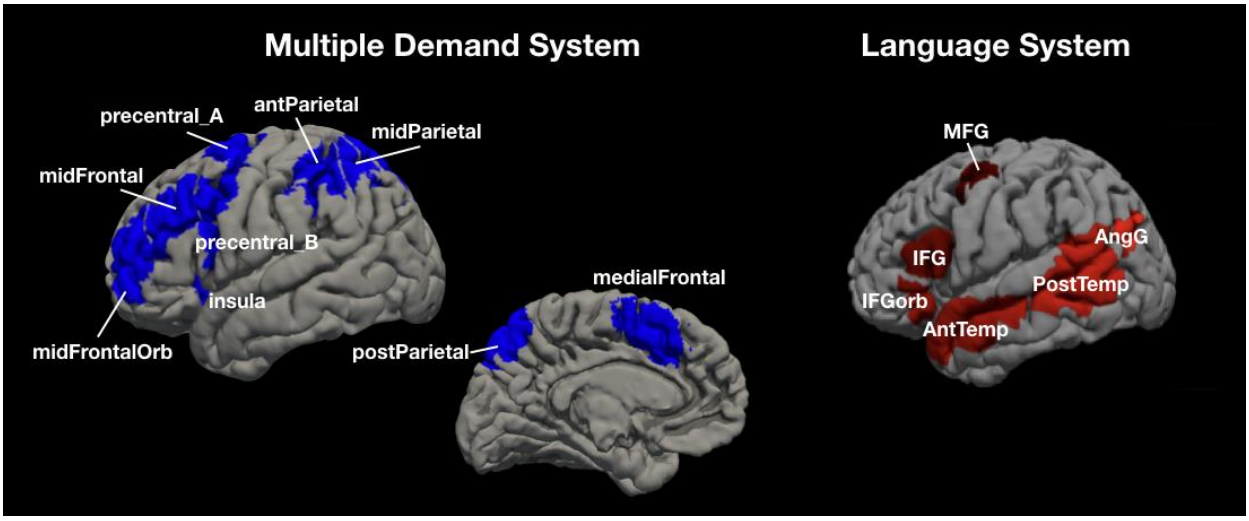

Figure S2. *The two candidate brain systems of interest. The areas shown represent the "parcels" derived from group-level representations of MD and language activity (NB: we show the left hemisphere parcels for the MD system, but the system is bilateral). These parcels are used to define the MD and language areas in individual participants. For each participant, the network of interest is comprised of the top 10% of voxels within each parcel with the highest t-value for the relevant contrast (MD - hard vs. easy spatial working memory task; language – sentence reading vs. nonword reading; see* ***Methods****). Abbreviations: mid – middle, ant – anterior, post – posterior, orb – orbital, MFG – middle frontal gyrus, IFG – inferior frontal gyrus, temp – temporal lobe, AngG – angular gyrus.*

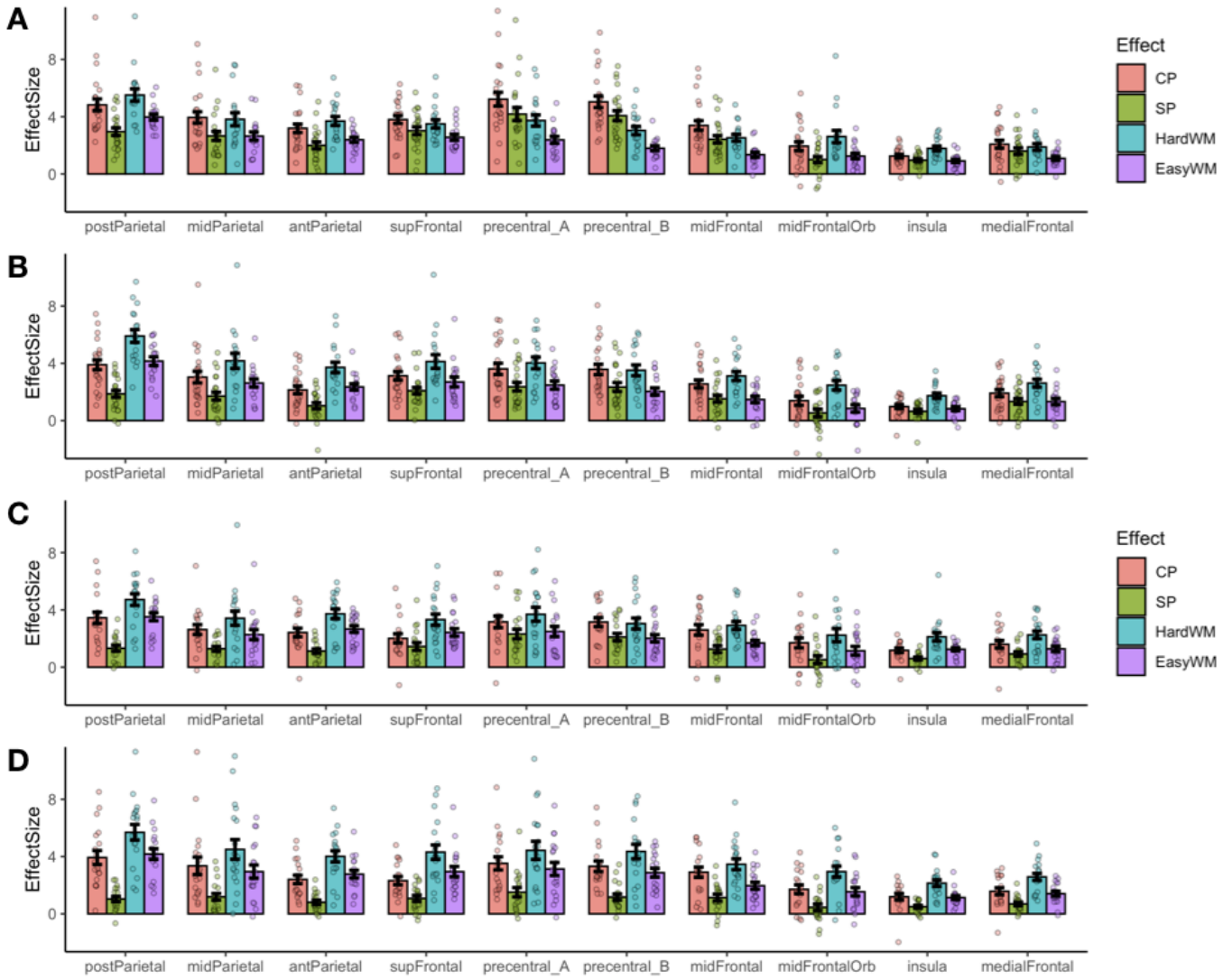

Figure S3. *ROI-level responses in the multiple demand system to the critical task (CP – code problems, SP – sentence problems) and the spatial working memory task (HardWM – hard working memory task, EasyWM – easy working memory task). (A) Experiment 1, Python; left hemisphere fROIs; (B) Experiment 1, Python; right hemisphere fROIs; (C) Experiment 2, ScratchJr; left hemisphere fROIs; (D) Experiment 2, ScratchJr; right hemisphere fROIs. No fROIs prefer both Python and ScratchJr code problems over the spatial working memory task.*

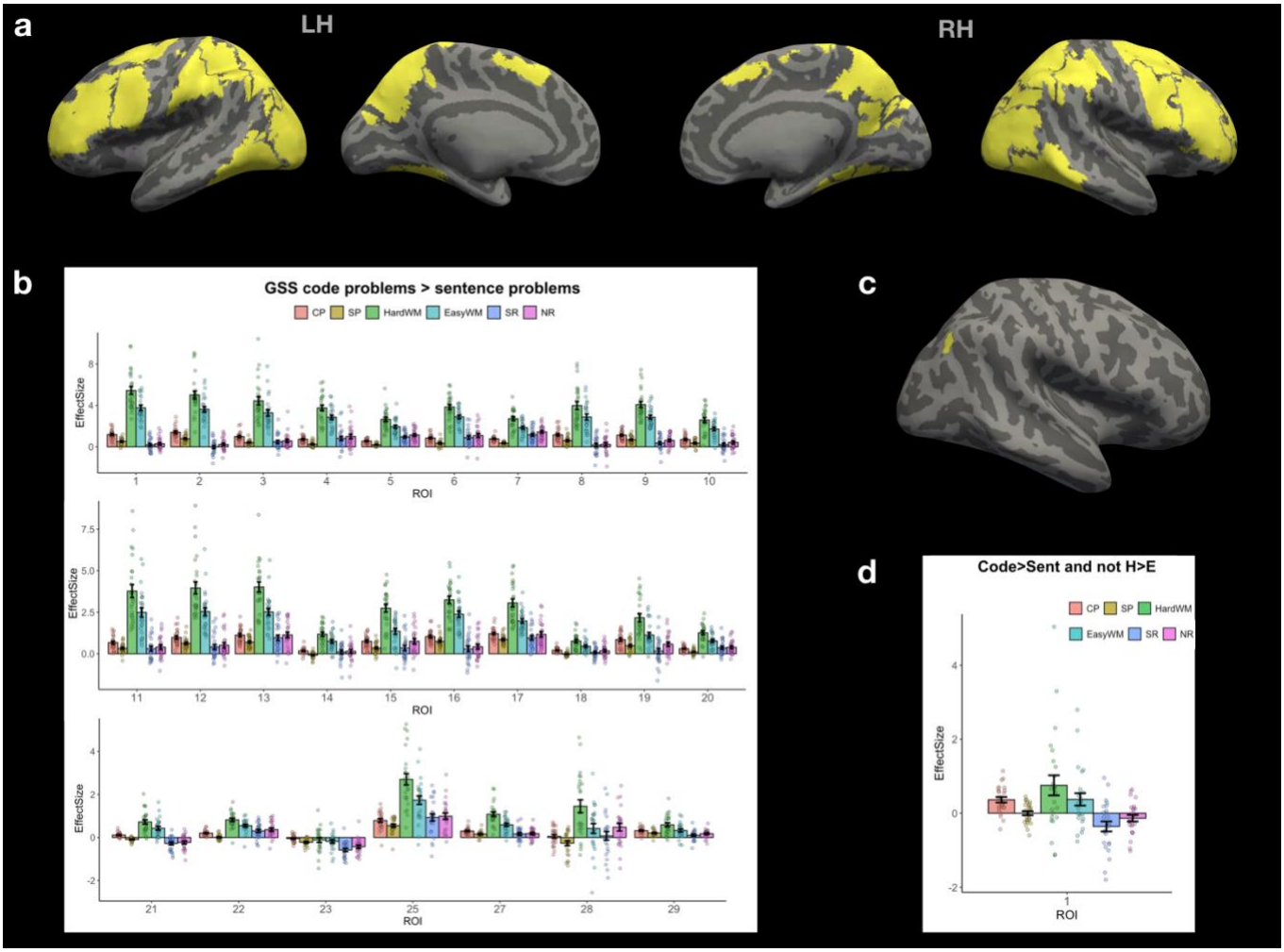

Figure S4. *Whole-brain group-constrained subject-specific analysis (GSS; Fedorenko et al, 2010) based on data from Experiment 1 shows the absence of code-only brain regions. (a) Parcels defined on the group level using the code problems > sentence problems contrast, p threshold 0.001, inter-subject overlap >= 70%. (b) Activation profile for the top 10% of voxels within each parcel in (a) across conditions. All code-sensitive regions exhibit high activity during the spatial working-memory task, suggesting that they belong to the MD system. (c) Parcels defined using the contrast above plus the "not hard working-memory task > easy working-memory task" contrast, p=0.5. Only one parcel was significant (right hemisphere). (d) Even that parcel's response profile shows high activity in response to the working-memory task, modulated by difficulty, rather than a code-specific response. Abbreviations: CP – code problems; SP – sentence problems; HardWM – hard working memory task; EasyWM – easy working memory task; SR – sentence reading; NR – nonword reading.*

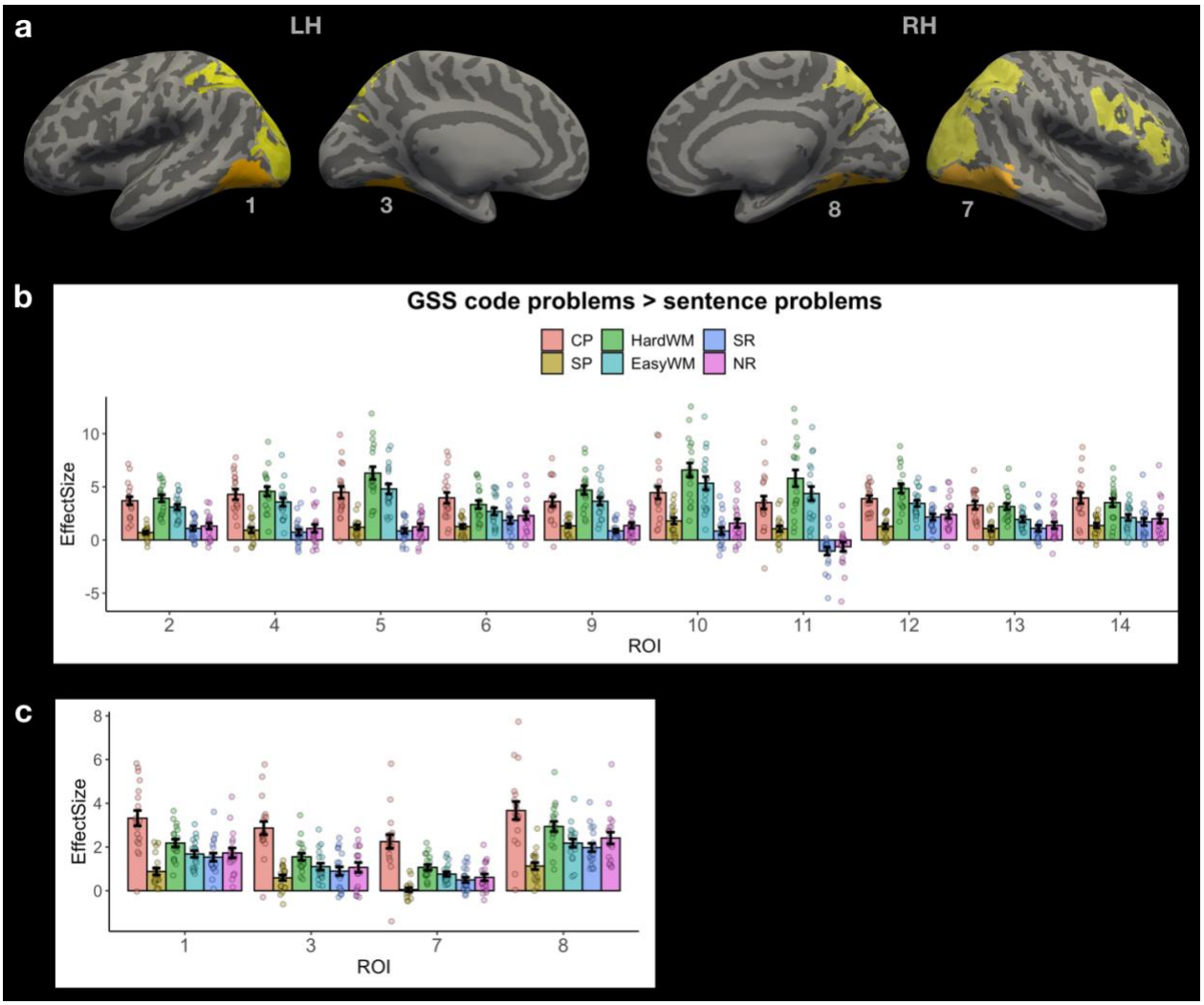

Figure S5. *Whole-brain group-constrained subject-specific analysis (GSS; Fedorenko et al, 2010) based on data from Experiment 2. (a) Parcels defined on the group level using the code problems > sentence problems contrast, p threshold 0.001, inter-subject overlap >= 70%. Parcels where the responses to ScratchJr code were stronger than responses to all other tasks are labeled and marked in orange; they include parts of early visual cortex and parts of the ventral visual stream. (b) Activation profile for the top 10% of voxels within each parcel in (a) marked in yellow. All regions exhibit high activity during the spatial working-memory task, suggesting that they belong to the MD system. (c) Activation profile for the top 10% of voxels within each parcel in (a) marked in orange. These fROIs exhibit higher responses to ScratchJr problems compared to a working memory task; given that they are located in the visual cortex, we can infer that they respond to low-level visual properties of ScratchJr code. A follow-up conjunction analysis using the contrast in (a) plus the "not hard working-memory task > easy working-memory task" contrast, p=0.5, revealed no significant parcels, indicating the lack of code-selective response. Abbreviations: CP – code problems; SP – sentence problems; HardWM – hard working memory task; EasyWM – easy working memory task; SR – sentence reading; NR – nonword reading.*

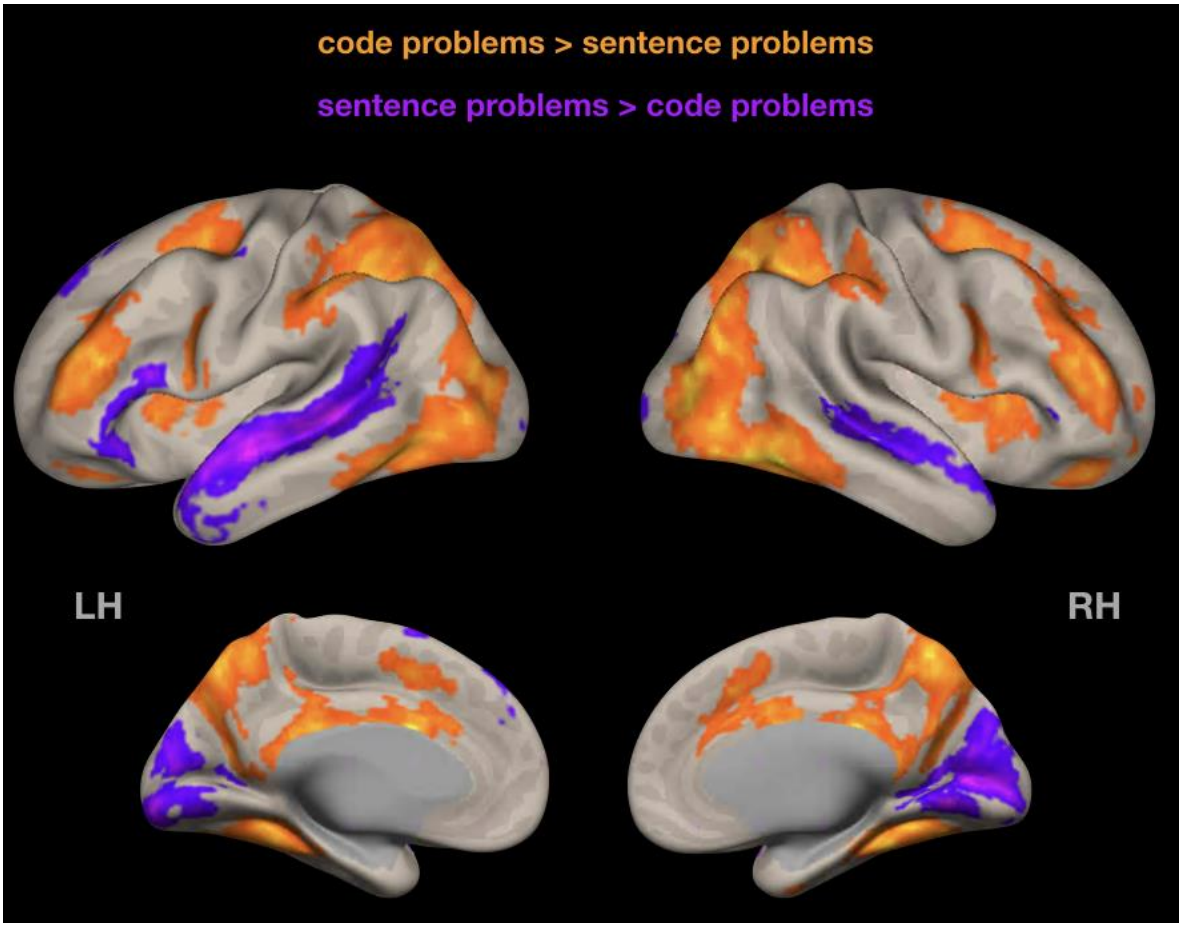

Figure S6. *Random-effects group-level analysis of Experiment 1 data (Python, code problems > sentence problems contrast). Similarly to analyses reported in the main text, code-evoked activity is bilateral and recruits fronto-parietal but not temporal regions. Cluster threshold $p < 0.05$, cluster-size FDR-corrected; voxel threshold: $p < 0.001$, uncorrected.*

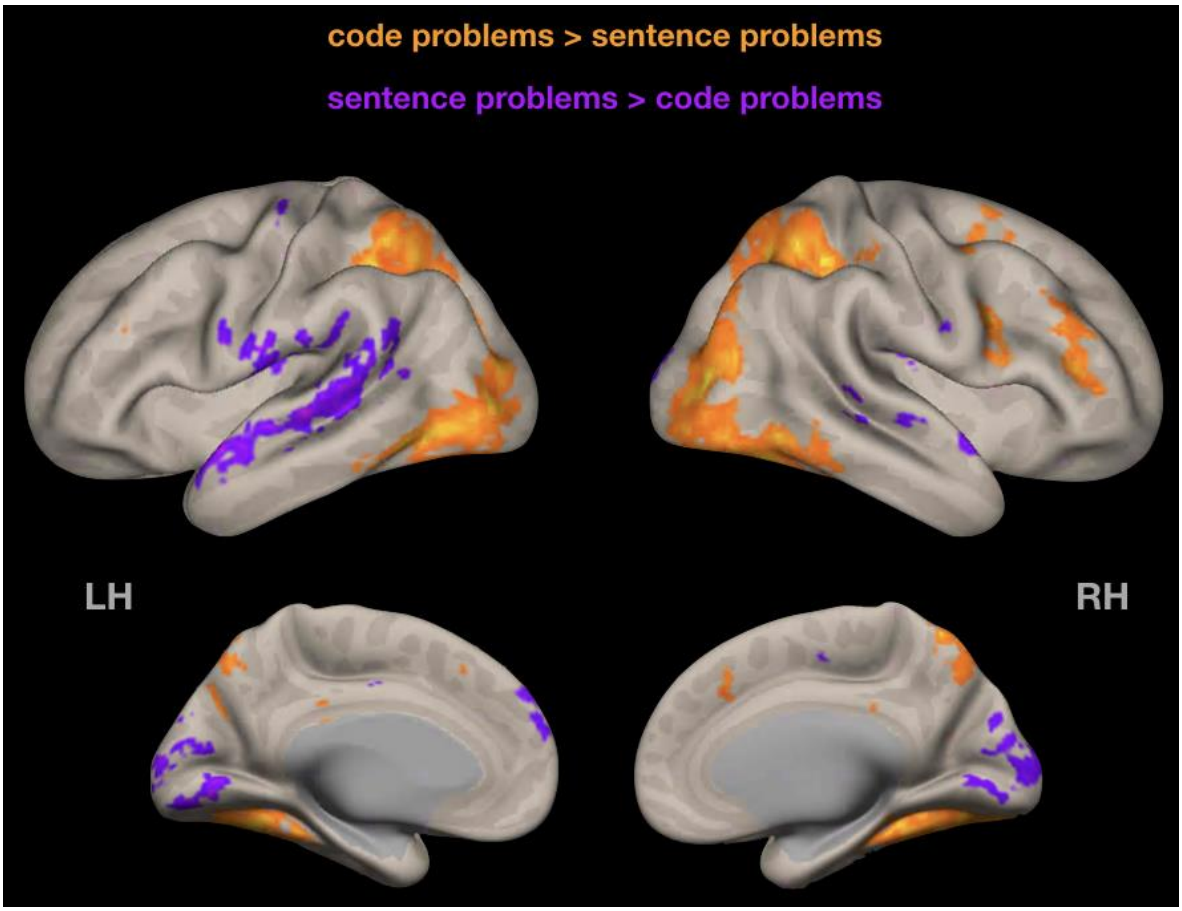

Figure S7. *Random-effects group-level analysis of Experiment 2 data (ScratchJr, code problems > sentence problems contrast). Similarly to analyses reported in the main text, ScratchJr-evoked activity has a small right hemisphere bias. Cluster threshold p < 0.05, cluster-size FDR-corrected; voxel threshold: p < 0.001, uncorrected.*

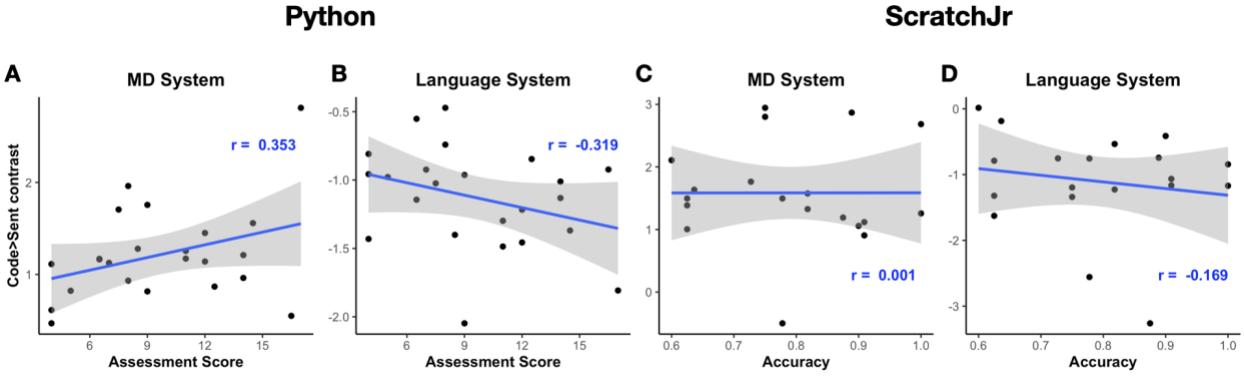

Figure S8. *The effect of programming expertise on code-specific response strength within the MD and language system in Experiment 1, Python (**A, B**) and Experiment 2, ScratchJr (**C, D**). Python expertise was evaluated with a separate one-hour-long Python assessment (see the paper's website https://github.com/ALFA-group/neural-program-comprehension); ScratchJr expertise was estimated with in-scanner response accuracies. No correlations were significant.*

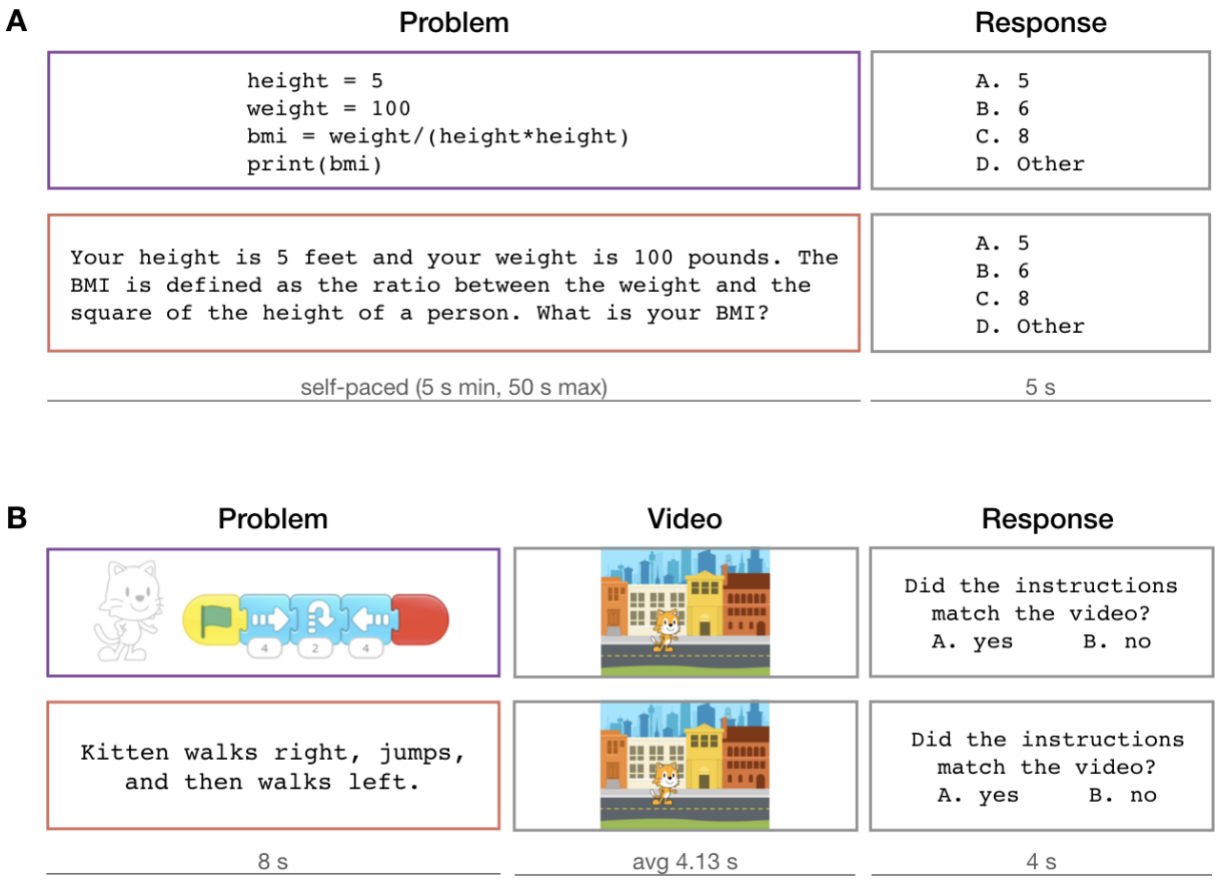

Figure S9. *Trial structure of the critical task. (A) Experiment 1 - Python (B) Experiment 2 - ScratchJr. All analyses use fMRI responses to the "problem" step.*

Table S1. *Responses to Python code problems (CP) vs. sentence problems (SP) and nonword reading (NR) in individual fROIs within the multiple demand system. P values are Bonferroni-corrected for the number of regions. Non-significant values are italicized and marked in gray.*

Hemisphere	ROI	Regression Term	Beta	p.value
L	precentral_A	Intercept	5.21	7.54E-13
		CP>SP	1.04	0.002
		CP>NR	3.36	4.11E-17
L	antParietal	Intercept	3.2	1.46E-15
		CP>SP	1.2	8.32E-07
		CP>NR	2.61	1.99E-17
L	medialFrontal	Intercept	2.1	2.10E-11
		CP>SP	0.48	0.214
		CP>NR	1.35	2.65E-08
L	supFrontal	Intercept	3.8	4.61E-18
		CP>SP	0.78	0.003
		CP>NR	2.96	7.43E-19
L	postParietal	Intercept	4.83	1.73E-20
		CP>SP	1.87	6.23E-07
		CP>NR	4.21	3.15E-18
L	midFrontal	Intercept	3.39	1.87E-16
		CP>SP	0.97	0.002
		CP>NR	2.51	3.50E-13
L	precentral_B	Intercept	5.01	1.85E-15
		CP>SP	0.97	0.001
		CP>NR	3.49	5.75E-19
L	midParietal	Intercept	3.95	1.33E-14
		CP>SP	1.29	3.79E-05
		CP>NR	3	1.70E-15
L	insula	Intercept	1.27	1.57E-15
		CP>SP	*0.29*	*0.158*
		CP>NR	0.67	1.19E-06
L	midFrontalOrb	Intercept	1.95	4.65E-09
		CP>SP	0.93	0.004
		CP>NR	1.05	7.50E-04
R	precentral_A	Intercept	3.62	7.23E-13
		CP>SP	1.25	3.80E-05
		CP>NR	2.14	6.00E-11
R	antParietal	Intercept	2.15	7.22E-13
		CP>SP	1.11	8.33E-05
		CP>NR	1.64	1.17E-08
R	medialFrontal	Intercept	1.94	5.44E-11
		CP>SP	0.59	0.006
		CP>NR	1.08	8.99E-08
R	supFrontal	Intercept	3.15	1.00E-16
		CP>SP	1.04	0.001
		CP>NR	2.53	6.31E-13
R	postParietal	Intercept	3.91	4.17E-18
		CP>SP	2.03	2.29E-10
		CP>NR	3.55	5.42E-19
R	midFrontal	Intercept	2.58	2.12E-14
		CP>SP	1.04	7.07E-04
		CP>NR	1.74	1.36E-08
R	precentral_B	Intercept	3.61	8.71E-14
		CP>SP	1.23	4.67E-05
		CP>NR	2.44	8.56E-13
R	midParietal	Intercept	3.04	8.23E-12
		CP>SP	1.33	1.51E-04
		CP>NR	2.16	2.43E-09
R	insula	Intercept	0.99	1.37E-09
		CP>SP	*0.33*	*0.18*

		CP>NR	0.52	0.002
R	midFrontalOrb	Intercept	1.42	2.81E-05
		CP>SP	0.86	0.012
		CP>NR	0.74	0.048

Table S2. *Responses to ScratchJr code problems (CP) vs. sentence problems (SP) and nonword reading (NR) in individual fROIs within the multiple demand system. P values are Bonferroni-corrected for the number of regions. Non-significant values are italicized.*

Hemisphere	ROI	Regression Term	Beta	p.value
L	postParietal	Intercept	3.45	2.76E-14
		CP>SP	2.13	1.88E-06
		CP>NR	2.65	1.33E-08
L	midParietal	Intercept	2.63	9.39E-13
		CP>SP	1.34	1.23E-04
		CP>NR	1.65	2.75E-06
L	antParietal	Intercept	2.42	2.42E-14
		CP>SP	1.3	0.001
		CP>NR	1.42	2.75E-04
L	supFrontal	Intercept	2.02	4.13E-07
		CP>SP	*0.57*	*1.173*
		CP>NR	1.14	0.007
L	precentral_A	Intercept	3.17	1.43E-07
		CP>SP	*0.85*	*1.207*
		CP>NR	*0.8*	*1.544*
L	precentral_B	Intercept	3.15	2.98E-10
		CP>SP	1.05	0.031
		CP>NR	1.33	0.002
L	midFrontal	Intercept	2.61	2.63E-09
		CP>SP	1.34	0.01
		CP>NR	1.5	0.003
L	midFrontalOrb	Intercept	1.7	1.23E-05
		CP>SP	1.18	0.017
		CP>NR	*0.89*	*0.182*
L	insula	Intercept	1.18	3.61E-07
		CP>SP	*0.58*	*0.161*
		CP>NR	*0.12*	*11.319*
L	medialFrontal	Intercept	1.6	9.49E-07
		CP>SP	*0.67*	*0.416*
		CP>NR	*0.47*	*2.048*
R	postParietal	Intercept	3.93	2.11E-16
		CP>SP	2.89	6.90E-07
		CP>NR	3.22	6.16E-08
R	midParietal	Intercept	3.35	1.75E-09
		CP>SP	2.17	2.12E-04
		CP>NR	1.83	0.002
R	antParietal	Intercept	2.4	1.43E-15
		CP>SP	1.59	1.67E-05
		CP>NR	1.53	3.58E-05
R	supFrontal	Intercept	2.32	1.95E-10
		CP>SP	1.24	0.007
		CP>NR	1.21	0.009
R	precentral_A	Intercept	3.52	8.22E-09
		CP>SP	2.02	2.26E-06
		CP>NR	1.27	0.004
R	precentral_B	Intercept	3.32	1.99E-10
		CP>SP	2.15	1.67E-07
		CP>NR	1.48	2.21E-04
R	midFrontal	Intercept	2.91	2.76E-11
		CP>SP	1.77	4.57E-05

		CP>NR	1.26	0.007
R	midFrontalOrb	Intercept	1.71	3.49E-06
		CP>SP	1.26	0.006
		CP>NR	*0.34*	*5.866*
R	insula	Intercept	1.19	1.02E-07
		CP>SP	*0.69*	*0.088*
		CP>NR	*0.2*	*7.779*
R	medialFrontal	Intercept	1.57	4.39E-08
		CP>SP	*0.89*	*0.064*
		CP>NR	*0.31*	*5.484*