

## **Using Student-Developed Narratives to Improve Learning and Engagement in Computer Problem-Solving Courses**

**Dr. Candido Cabo, New York City College of Technology**

Candido Cabo is a Professor in the Department of Computer Systems Technology at New York City College of Technology, City University of New York (CUNY). He earned the degree of Ingeniero Superior de Telecomunicacion from the Universidad Politecnica de Madrid (Spain) in 1982, and a Ph.D. in Biomedical Engineering from Duke University (Durham, NC) in 1992. He was a post-doctoral fellow at Upstate Medical Center, State University of New York (Syracuse, NY), and a research scientist in the Department of Pharmacology at the College of Physicians and Surgeons of Columbia University (New York, NY). Since 2005, he has been a member of the doctoral faculty at the CUNY Graduate Center. His research interests include computer science and engineering education and the use of computational models to understand and solve problems in biology.

**Dr. Reneta Davina Lansiquot, New York City College of Technology**

Reneta D. Lansiquot is Associate Professor and Program Director of the Bachelor of Science in Professional and Technical Writing at New York City College of Technology of the City University of New York, where she earned an AAS in Computer Information Systems and a BTech in Computer Systems. She earned a MS in Integrated Digital Media at Polytechnic University and a PhD in Educational Communication and Technology at New York University. Her research focuses on interdisciplinary studies. Her first book is entitled *Cases on Interdisciplinary Research Trends in Science, Technology, Engineering, and Mathematics: Studies on Urban Classrooms* (Information Science Reference, 2013). Her two forthcoming books are entitled *Interdisciplinary Pedagogy for STEM: A Collaborative Case Study* (Palgrave Macmillan) and *Technology, Theory, and Practice in Interdisciplinary STEM Programs: Connecting STEM and Non-STEM Approaches* (Palgrave Macmillan).

# Using Student-Developed Narratives to Improve Learning and Engagement in Computer Problem-Solving Courses

## Abstract

In our Computer Systems major, we require all students to take a problem-solving course (PS) to prepare them for subsequent courses in computer programming. As part of the PS course, students use flowchart interpreters to learn basic procedural programming concepts. Students often have difficulty translating word problems into flowcharts because they do not have a good understanding of the problem's domain. To write an effective computer program to solve a problem, programmers must first understand the problem.<sup>1,6,14,15</sup> Thus, the student's inability to conceive of the problem intellectually may prevent them from creating an effective solution.

The goal of our project was to create narratives (case-studies) that students could understand, relate to, and be engaged with, so they can be used as the contexts to learn problem-solving and procedural programming skills in the flowcharting component of the PS course. Since students know what is relevant to them, we selected a group of five students majoring in Computer Systems to help us develop those narratives.

The use of student-developed narratives affected performance in different flowcharting structures (sequence, selection, repetition) differently. Overall, the data suggests that the use of case studies was beneficial for increasing performance in flowcharting assessments using selection structures, modestly beneficial in assessments using repetition structures, and of no benefit in assessments using just sequence. Despite the learning benefit, a majority of students and instructors were resistant to using case-studies in this course and thought that reading stories does not belong in a problem-solving/computer programming course. Therefore, novel approaches to engage students and instructors in interdisciplinary thinking and practice are necessary.

## 1. Student Stories as Contexts for Problem-Solving and Computer Programming Courses

The PS course is required for all Computer Systems first-year majors, but it is also taken by students with other majors. In the first part of the PS course, students learn basic procedural programming concepts, like input, sequencing, selection (if/else), repetition, and output using flowchart interpreters, like Visual Logic ([www.visuallogic.org](http://www.visuallogic.org)). When solving flowcharting problems, students have difficulty translating word problems into computer algorithms. Moreover, most problems proposed to students are closely related to mathematics and accounting, and our students are not well prepared in mathematics. Students are neither interested nor engaged by the flowcharting problems. Understanding and engaging the problem domain to be solved by implementing a computer program should be a prerequisite for writing the computer program itself.<sup>1,6,14,15</sup> Therefore, the students' inability to create a mental model<sup>8</sup> of a given problem domain hinders their ability to develop problem-solving skills and write computer programs.

In the second part of the PS course, students use *Alice*, a three-dimensional animation software environment ([www.alice.org](http://www.alice.org)),<sup>4,5</sup> to create interactive animations, in order to further develop and reinforce their problem-solving and procedural programming skills as well as to introduce object-oriented programming concepts and skills. We have shown that, when students develop engaging narratives which they can then implement as a computer program using *Alice*, their success in the PS course improves.<sup>2,3,10,11,12,13</sup> However, an important difference between the earlier part of the course and the later part is that, in the flowcharting component, instructors generally pose the problems that the students will solve to develop their problem-solving skills. That is not the case in the *Alice* component of the course because students create their own stories (i.e., their own problems) that they then solve by implementing a video game prototype.

Here, we describe a solution to the problem of student engagement by creating problem domains that students could understand, relate to and be engaged with, so they can be used as the contexts to develop problem-solving and procedural programming skills in the flowcharting component of the PS course. Our approach is based on the premise that students themselves know better which problems are relevant to them, which problems they can relate to and understand. In fall 2014, we selected a group of five students, majoring in Computer Systems, and who passed the PS course within the last three years, to develop stories that could be used as context to solve flowcharting problems. Those five students took a section of the PS course that was linked to a learning community (LC),<sup>9</sup> so, in addition to understanding well the course in which the stories were going to be used, they had considerable narrative and writing skills. The students themselves suggested which flowcharting assignments could be assigned with the stories. The stories were developed iteratively, following a combination of individual writing, group discussion, and faculty suggestions, to further improve the versions of the stories. Students were provided with a small stipend.

The five stories developed by the former LC PS students (3 male and 2 female) range in length from 1,200-2,600 words, and have different themes. The themes reveal what is important, interesting, and engaging to our students. One story is a fantasy (*Shell Sleuth*), another is science fiction (*Mind Games*), and the other three are New York City stories imbued with magic realism (*Belvedere*, *Movement in Green*, and *The Shadows of Invisibility*).<sup>9</sup> Here are the plots of their stories:

### *Shell Sleuth*

This story takes place in a fantasy world during a war between Elves and Orcs. The protagonist is an Elven spy named Zerk who must calculate the size of the enemy army. To do this, he has disguised himself to sneak aboard one of the enemy creature ships. He is on his way to an outpost to which all the forces are heading for a celebration. Once there, he will be able to calculate the size of the enemy forces by counting the ships at the post.

### *Mind Games*

What happens when you create a machine with artificial intelligence? A better future or total annihilation? Well, the genius Professor Harry Odum didn't think much of his invention ARIIA. He also did not think much of placing ARIIA at the helm of the United States nuclear missiles. When the professor's son, Michael, comes to visit the Arizona military base, ARIIA begins to act strangely. Michael initiates a game with ARIIA with all of the military's nuclear missiles as the

prize for the winner and total destruction for the loser. Join Michael as he becomes entwined in a game of machine versus human with a machine that knows a little too much about Michael.

### *Belvedere*

Isaiah Jones is experiencing the American dream, but not one filled with glamour, expensive cars, and a college education. Isaiah is experiencing the “other” American dream, days of repetitive and monotonous work, no college degree, and no money. “Having no luck” cannot even begin to describe Isaiah’s life, but he is not a lost cause because he possesses an impressive witty brain. When hope seems lost and there seems to be no chance of a better life, James Hensen, a rich entrepreneur, offers an opportunity to the young man in the form of delivering packages for large sums of money. However, James does not seem to be the entrepreneur he presents himself to be; rather, Isaiah grows suspicious when James asks him to meet at Belvedere Castle in Central Park. Isaiah, presented with the quandary of delivering the packages or not, begins to weigh his options. What will he choose to do?

### *Movement in Green*

Ivory Nyr, an ordinary college girl, discovers only three nights before New Year’s Eve that a mysterious, invisible, omnipotent force named Magister has been watching her. After Ivory completes a Good Samaritan deed, Magister assumes human form to speak with her directly on an empty, speeding train that seems to never reach its destination. Here, the entity reveals Ivory has already participated in three challenges as part of his “game” and informs her that, with each challenge, a life is saved. Ivory has a mere 72 hours to complete the remaining challenges. If she fails to fulfill them before the New Year arrives, or if she declines the offer, the lives at stake will be lost—including her own.

### *The Shadow of Invisibility*

Samantha, a first-year college student, starts as an ordinary girl who often feels unnoticed. Her day begins with its usual boring pattern, but something about this day feels strange. She senses some mysterious dark figures are following her. On her way home, several of them confront her and reveal themselves to be an underground family. They tell Samantha she has the power of invisibility and ask her to join them. Although she declines, her life is forever changed. She becomes extremely sneaky and devious towards the people around her. Ultimately, Samantha’s life is forever changed.

## **2. Implementation of the Case Studies**

To implement the case studies, we conducted a workshop with five instructors who would teach the PS course both in Fall 2014 and Spring 2015 to share suggestions on how to incorporate those stories in the curriculum. The stories were used as the context to create flowcharting assignments, some suggested by the students who wrote the case studies and others by the instructors, for in-class or outside-class assignments, with students preferably working in teams. To quantify the effect of the case-studies on student’s ability to solve-problems algorithmically using flowcharts, all instructors agreed on a set of 10 common questions involving sequencing, selection (if/else), and repetition loops (see Appendix 1). In Fall 2014, the 10 common questions were used to evaluate the baseline (or control) student performance (n=182). In Spring 2015, we

piloted *Shell Sleuth*, *Mind Games*, and *Belvedere* (see Appendix 2) with 202 students in 9 PS course sections; to evaluate the effect of the use of case-studies on student performance on flowcharting, students were assessed with the same 10 common questions. Instructors were given guidelines to help them implement the case-studies in their respective sections of the PS course (see Appendix 3). All five instructors taught PS sections of the course in fall 2014 and spring 2015. To prevent the variability that comes with instructor grading and case study implementation styles, we compared student performance during control and case studies for each individual instructor.

### 3. Effect of the Case Studies on Student Performance on Flowcharting

Figure 1 shows the percent of students with adequate performance ( $\geq 70\%$ ) in the different assessments during control (empty bars) and when case studies were used (gray bars) for the different instructors. Also shown is the average for all instructors. Note the variability for the different instructors during control: For instructor #2, about 50% of the students showed adequate performance, while for instructors #3 and #4, about 90% of the students showed adequate performance. The percent of students with adequate performance increased for 4 instructors by values ranging from 3%-13%, but it decreased for instructor #1 by 11% (see also Table 1). The data shows that there is instructor variability, not only in the control assessments, but also in the effect of student performance when using case studies.

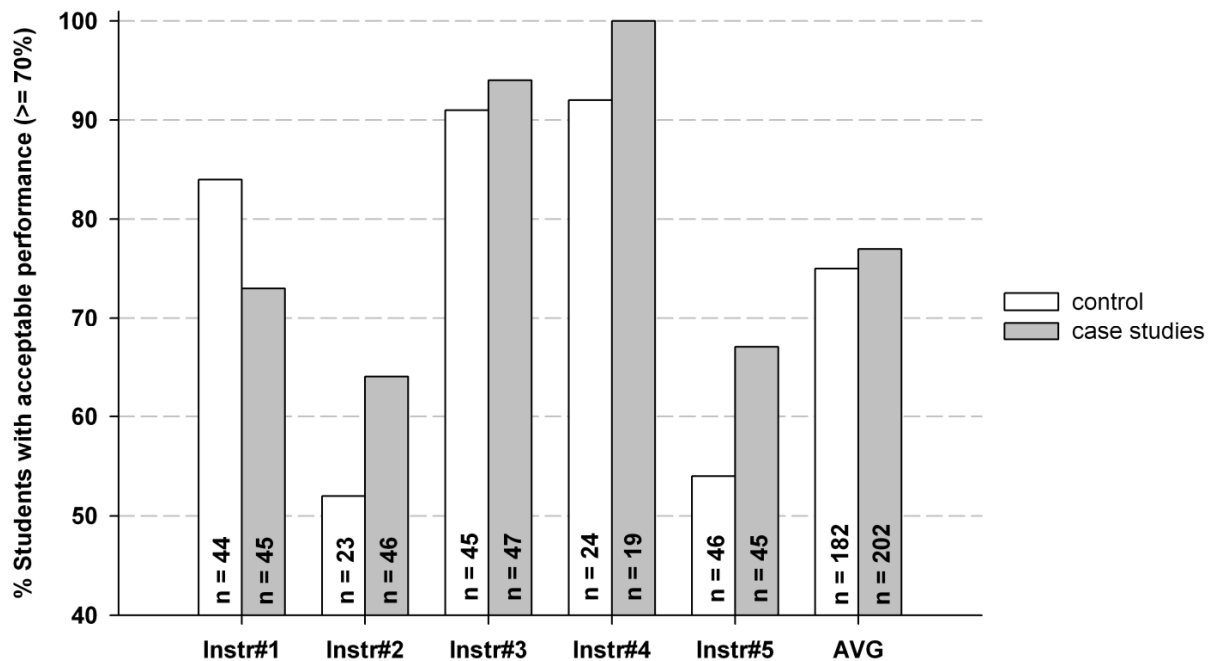


Figure 1. Percentage of students with an acceptable performance (70% or better) in the 10 common questions in Fall 2014 (control; n=182 students) and in Spring 2015 (case studies; n=202 students) for five instructors.

Table 1 shows the percent increase of students with an acceptable performance ( $\geq 70\%$ ) when using case studies for 5 different instructors and 3 different flowcharting assessments (sequence, selection, and repetition). Performance in assessments using only flowcharting sequences was not changed by the use of case studies: There was a decrease for 2 instructors, no change for one and an increase for the other 2 instructors. For 4 of the 5 instructors, performance in the flowcharting selection assessments increased between 5%-26%; for the other instructor, there was a decrease of 15%. For 3 of the 5 instructors, there was an increase in performance in repetition assessments from 1%-12%; for the remaining instructors, there was no change for one and a decrease of 3% for the other. So, overall the data suggests that the use of case studies was beneficial for increasing performance in selection assessments, modestly beneficial for repetition assessments, and of no benefit for sequence assessments.

	Sequence	Selection	Repetition	Average
<b>Instructor #1</b>	-11%	-15%	+1%	-11%
<b>Instructor #2</b>	-4%	+26%	+4%	+13%
<b>Instructor #3</b>	0%	+5%	0%	+3%
<b>Instructor #4</b>	+5%	+18%	-3%	+8%
<b>Instructor #5</b>	+6%	+6%	+12%	+12%

*Table 1.* Percent increase of students with an acceptable performance (70% or better) when using case studies for 5 different instructors and 3 different flowcharting assessments (sequence, selection, and repetition). Red numbers indicate that the percentage of students with acceptable performance decreased with the use of the case studies.

#### 4. Student and Faculty Opinion on the Case-studies

For an opinion survey about case studies, 141 students responded (Figure 2). Overwhelmingly, the stories that students liked the most were *Mind Games* (48%) and *Belvedere* (25%). About 9% of students did not like any of the stories. Students gave mixed responses to statements like, “The case-studies held my attention” and “The case-studies helped me understand how to create flowcharts,” and the question, “Would you like to see more case studies in this course?” About 60% of students agreed or strongly agreed that the cases were interesting, held their attention, and were useful in learning flowcharting. Still, a majority of students (53%) would not like to use more case-studies in this course. Their explanations varied, but the majority of students thought that the stories were too long and that reading stories does not belong in a problem-solving with computer programming class.

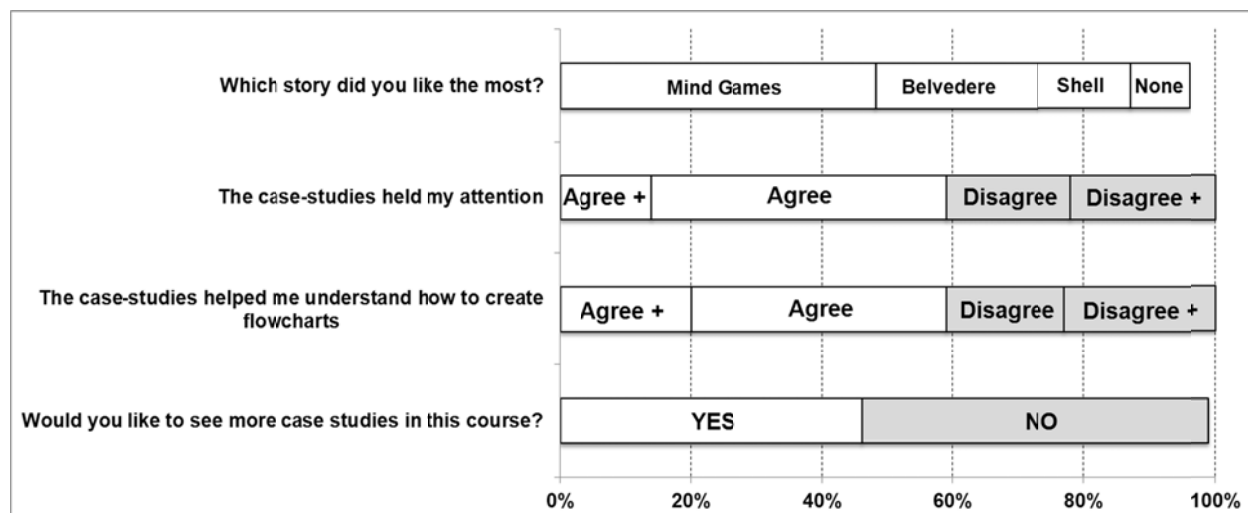


Figure 2. Summary of responses to an opinion survey about the case studies by 141 students.

Whether students thought the case studies were useful or not varied for the different instructors (Table 2). Positive student reactions to the case studies ranged from 19% (instructor #3) to 86% (instructor #4). The data in Table 2 seems to indicate a degree of correlation between the responses by students and instructors on the usefulness of the case studies for the course. Three instructors (#2, #3, and #5) thought that the case-studies were not a useful addition to the PS course, whereas 2 instructors thought it was (#1 and #4). Students in sections taught by Instructor #3 (who did not like using case-studies in this course), overwhelmingly did not want to see more case studies in this course (81%). Students in sections taught by instructor #4 (who liked using case studies in this course), overwhelmingly wanted to see more case studies in the course (86%).

It should also be noted that student performance is not correlated with whether students or faculty thought the case studies were useful or not. For example, neither Instructor #2 nor the majority of students in the instructor's sections (66%) liked to use case studies, but the average performance was increased by 13% when compared to control. Even for Instructor #3 (who did not like to use case studies), whose students had an overwhelming reaction against using case studies (only 19% like to use case studies), the use of case studies modestly increased performance. Unfortunately, the student survey data for the sections taught by Instructor #1, whose sections showed a decrease in student performance, is not available.

	Instructors who thought case studies were useful	% of students who thought case studies were useful	Average increase in performance
<b>Instructor #1</b>	Yes	-	-11%
<b>Instructor #2</b>	No	44%	+13%
<b>Instructor #3</b>	No	19%	+3%
<b>Instructor #4</b>	Yes	86%	+8%
<b>Instructor #5</b>	No	66%	+12%

Table 2. Students and faculty opinion on the use of case studies. Student responses and performance were segregated by instructor. Student responses in the sections taught by instructor #1 were not available.

## 5. Next Steps

Due to the overall positive effect of the case studies on student flowcharting performance, we will continue implementing case studies. We will revise the case studies, taking into account student and faculty opinions. This should include, for example, shortening *Mind Games*, which uses all basic procedural programming concepts—such as input, sequencing, selection (if/else), repetition (for and while loops), and output—as well as including additional student-created stories in the preferred genres: science fiction and magical realism. We will aim to have these new stories not exceed 1,500 words. This shorter length should allow for more seamless integration into current lesson plans.

Instructor buy-in is also critical to the success of any pedagogy, including the use of case studies. We will present also our findings to the department, targeting faculty who teach the PS course. In order to encourage more instructors teaching the PS course to incorporate case studies in flowcharting examples, we will show them the positive effect of the case studies on student performance. Additionally, it will be important to create professional development workshops for instructors to understand the importance of engaging in interdisciplinary thinking and practices.

## 6. Conclusions

Building on the success of our programming narratives LC,<sup>2</sup> we used student-developed stories to serve as contexts for computer programming courses. This intentional interdisciplinary approach to problem solving suggests that the use of case studies was beneficial for increasing performance in selection assessments, modestly beneficial for repetition assessments, and of no benefit for sequence assessments. Despite the overall learning benefit, scaling interdisciplinary approaches in the classroom remains a challenge: a majority of students and instructors were resistant to use case studies in this course because they believed that reading stories does not belong in a problem-solving/computer programming class. Novel approaches to engage students and instructors in interdisciplinary thinking and practice are necessary, both to improve student performance in courses and to help students develop the skills they will need to be effective leaders facing the complex problems of the twenty-first century.

## 7. References

- [1] Brooks, R. E. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies*, 9, 737-751.
- [2] Cabo, C., & Lansiquot, R. D. (2014). Synergies between writing stories and writing programs in problem-solving courses. In *Proceedings of the 2014 IEEE Frontiers in Education Conference* (pp. 888-896). New York: IEEE.
- [3] Cabo, C., & Lansiquot, R. D. (2013). Development of interdisciplinary problem-solving strategies through games and computer simulations. In R. D. Lansiquot (Ed.) *Cases on interdisciplinary research trends in science, technology, engineering, and mathematics: Studies on urban classrooms* (pp. 268-294). New York: Information Science Reference.



- [4] Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15(5), 107-116.
- [5] Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3D graphics to prepare novices for CS1. *Computer Science Education*, 13(1), 3-30.
- [6] Davies, S. P. (1993). Models and theories of programming strategy. *International Journal of Man-Machine Studies*, 39(2), 237-267.
- [7] Kuh, G. D. (2008). *High-impact educational practices*. Washington, D.C.: AAC&U.
- [8] Johnson-Laird, P. N. (1983). *Mental models*. Cambridge: Cambridge University Press.
- [9] Lansiquot, R. D., & Cabo, C. (2015). Strategies to integrate writing in problem-solving courses: Promoting learning transfer in an interdisciplinary context. In *Proceedings of the 122<sup>nd</sup> American Society for Engineering Education Annual Conference*. Washington, DC: ASEE.
- [10] Lansiquot, R., & Cabo, C. (2014). Strengthening the narrative of computing with learning communities. In *Proceedings of World Conference on Educational Media and Technology 2014*. Chesapeake, VA: AACE.
- [11] Lansiquot, R. D., & Cabo, C. (2011). Alice's adventures in programming narratives. In C. Wankel & R. Hinrichs (Eds.), *Transforming virtual learning: Cutting-edge technologies in higher education* (Vol. 4, pp. 311-331). Bingley, UK: Emerald.
- [12] Lansiquot, R., & Cabo, C. (2010). The narrative of computing. In *Proceedings of World Conference on Educational Media and Technology 2010* (pp. 3655-3660). Chesapeake, VA: AACE.
- [13] Lansiquot, R. D., Satyanarayana, A., & Cabo, C. (2014). Using interdisciplinary game-based learning to develop problem solving and writing skills. In *Proceedings of the 121<sup>st</sup> ASEE Annual Conference*. Washington, DC: ASEE.
- [14] Rist, R. S. (1995). Program structure and design. *Cognitive Science*, 19, 507-562.
- [15] Spohrer, J. C., Soloway, E., & Pope, E. (1989). A goal/plan analysis of buggy Pascal programs. In E. Soloway and J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 355-399). Hillsdale, NJ: Lawrence Erlbaum.

## **Acknowledgements**

We would like thank students Solanlly Hernandez, Tykila McCray, Noel Melendez, Andy Persaud, and Walter Rada for developing the stories that serve as the context of the case studies. We would also like to thank instructors Michael Gomez, Jeremy Graham, Mary Halsey, Andrea Lawrence, and Roger Wieber for implementing the case studies in their sections of the PS course. We also recognize Perkins Major Effort Component project, which is supported by the New York State Education Department (NYSED) Grant Number 51980-05-24.

## **APPENDIX 1: Ten questions used for assessment of students' ability to solve problems algorithmically using a flowchart.**

### **Sequence:**

- 1) Using Visual Logic solve the following problem: Given the base and the height of a triangle calculate its area. The output should show the base and height entered by the user and the area calculated by the program. (Input: 2 points; Processing: 6 points; Output: 2 points)
- 2) Using Visual Logic input two numbers. Then calculate the sum, the difference, the product, the quotient, and the average. Output each of those values in a separate line. (Input: 2 points; Processing: 5 points; Output: 3 points)
- 3) Create a Visual Logic flowchart that calculates the number of calories in a menu item. The flowchart should take four inputs: the name of the menu item, the number of grams of fat, the number of grams of protein, and the number of grams of carbohydrates the item contains. The flowchart then calculates and reports the total number of calories in the item, as well as the percent of calories that come from fat. Use the following information to perform the calculations: 1 gram fat = 9 calories 1 gram carbohydrate = 4 calories 1 gram protein = 4 calories (Input: 2 points; Processing: 6 points; Output: 2 points)

### **Selection:**

- 4) When you go to a restaurant you need to calculate the tip that you need to give the waiter. Write a flowchart that allows a user to enter the total bill, and whether the service was very good or just fine. If the service was very good, the tip will be 25% of the total bill. If the service is just fine the tip will be 15% of the total bill. Calculate the tip (Input: 2 points; Processing: 6 points; Output: 2 points)
- 5) Create a flowchart that requests the points scored by the B'klyn Nets and the Lakers at the end of regular time. The program should display "B'klyn Nets won!" if the Nets won, or "Lakers won!" if the Lakers won, or "Overtime needed" if there was a tie. (Input: 2 points; Processing: 6 points; Output: 2 points)
- 6) Create a flowchart that converts a numeric grade to a letter grade. The user should be able to enter a numeric grade from 0 to 100. The flowchart will calculate the letter grade based on:  $A \geq 90$ ;  $80 \leq B < 90$ ;  $70 \leq C < 80$ ;  $60 \leq D < 70$ ;  $F < 60$  (Input: 2 points; Processing: 6 points; Output: 2 points)

### **Repetition (FOR):**

- 7) Create a flowchart to calculate the squares of the first 10 even numbers (starting with 0). The output should show the number and its square. (Input: 2 points; Processing: 6 points; Output: 2 points)

- 8) Let's make a deal. For one month (30 days), I give you \$1000 dollars every day if you give me 1 penny the first day, and then every day you double the amount of pennies I have. Calculate how much money you and I will get at the end of the 30 days. (Input: 1point; Processing: 8 points; Output: 1 point)

**Repetition (WHILE):**

- 9) You take out an interest free car loan for \$10,000 and, to repay the loan, you have to pay \$150/week. Create a flowchart using a WHILE loop to calculate the balance due each week until the loan is paid. (Input: 1point; Processing: 8 points; Output: 1 point)
- 10) Create a flowchart such that you input a number between 1 and 100. Then the user needs to guess the number. If the user's guess is higher than the number, the program should display "Too high, try again." If the user's guess is lower than the number, the program should display "Too low, try again." When the user guesses the number, the program should display a congratulations message and the number of guesses that the user had to make to guess the number. (Input: 1point; Processing: 8 points; Output: 1 point)

## APPENDIX 2: Example of a case study

### Belvedere Solannly Hernandez

The Big Apple, New York City, one of the most famous cities in the world. Also, one of the most over-populated, over-congested and over-represented city in the world. The history of New York City is largely based on stories of success, triumph and glory. Unfortunately, that only accounts for a small number of the population while the truth of it is that not all stories have happy endings.

Isaiah Jones, a 21-year-old who is half-Black and half-Puerto Rican, is a prime example of a New York City resident. He has lived all of his life in the Bronx, one of the five boroughs of New York City, and is still waiting to get his share of success. He attended Bronx Community College for two semesters and had to drop out in order to help out at home. He started working the night shift as a dishwasher at a restaurant in downtown Manhattan, making \$15 an hour and working forty- and fifty-hour shifts each week. Life was becoming monotonous, repetitive and tiring. Isaiah's day consisted of waking up at one in the afternoon to catch the 2 p.m. D train, working until 2 a.m. and going out to bars to de-stress until 6 a.m. and getting home around 8 a.m. to sleep, only to wake up and do the same thing over again.

Life for Isaiah had not been easy; growing up in a poor community did not help. The struggles and hardships were a cycle of intense displeasure and sadness stemming from problems from the humiliation of living in public housing, not being able to afford rent, relying on food stamps to get by every month and his mother's deteriorating health.

Isaiah had an upper hand, though: he was smart, he was full of wit and he never backed away from a challenge. He had become an expert in making difficult decisions that would change the course of his life. James Hensen, a rich white man he met on one of his late night deliveries—because, of course, the job title of dish washer was not only limited to washing dishes—was an example of the different type of people that can be found in New York City. James was an entrepreneur; he was starting a new business and was searching for the right set of employees to hire. James was looking for ten strong men that would carry out his business; however, they not only needed to be strong but they also had to have a strong personality. Isaiah was the right employee for James. On the night they met, James asked Isaiah a couple of questions.

“So you are from the Bronx? I've heard many things about the people from the Bronx. Are they true?”

“Well, it all depends, man. What have you heard? If you think we're all the same, well that's not true.”

“Hmm... by the same, what do you mean?”

“Like you know, you people think everyone from the Bronx is bad, like we just all drug dealers and stuff, but we're not. At least I'm not. ”

“We people.” James looks Isaiah up and down. “Why don't you come to my office tomorrow morning, 8 a.m.? I'd like to talk to you about a job.”

“I don't need a job. I have a job and I ain't looking for one from a stranger either,” says Isaiah, interrupting him.

“But of course you're not, Isaiah...” James takes out a business card from his pocket and hands it over to Isaiah.

Isaiah takes the card out of courtesy and leaves before James could say another word. He returns to the restaurant and finishes his shift. Later, as Isaiah waits at the platform for more than forty-five minutes, his mind moves on the strange man he met. He takes out the business card that James had given him and looks at it. The address is somewhere in Central Park—Belvedere Castle. Isaiah starts to laugh; for a second there, he actually believed that James was a business man offering him a real job. The train finally arrives and he hops on. He gets home and sets his alarm to seven; even though he will only have three hours of sleep, he wants to go check out this castle; it has sparked his curiosity.

Isaiah is up, bright and early. He had forgotten how beautiful Central Park could be in the beginning of fall. The leaves were changing colors and, since it was early in the morning, the park was practically empty. He heads over to Belvedere Castle and, to his surprise, he finds James there.

“Ah, Mr. Isaiah, so you took my offer.”

“Ok, so what’s the deal? What are you doing in this castle? It’s Central Park, dude.”

“The real question is what are *you* doing here?” James stares at Isaiah.

Isaiah stands quietly. He does not know what to say, unsure why he has come to the castle; all he knows is that he is curious about this man but he cannot explain why. He looks at James and notices that he is all dressed up, suit and tie, as if he actually had a business. Isaiah is confused.

“So, why don’t you tell me a bit about yourself Isaiah? Or maybe why don’t I tell you a bit about myself.” James squints and looks at Isaiah. He breathes in deeply and begins to talk. “I am from New York, born and raised; my business has been in the family for the past ehh... century or so. But unfortunately I am the only one left and since the business must continue, I am looking for ten strong men to help me out.”

“Why ten?”

“Ehh... good question, Mr. Isaiah. You see how everyone has a lucky number. Well, my family’s lucky number is ten. It’s just tradition; don’t want to break tradition. OK. Mr. Isaiah, first things first. This castle here, Belvedere means ‘beautiful view’; we have a beautiful view of Central Park. I need you to work for me. You have what I need and I want it. Now, here is the catch. All you have to do is take my offer and you have a job.”

“Knew there was a catch. Good knowing you, James...”

“But don’t you want to hear me out? Come on. Don’t you have that tuition bill you have to pay to . . . uhh . . . What’s that school? Oh yeah Bronx Community College, oh and those ambulance bills. Poor ma. She gets sick all the time and her insurance does not cover that. What about your little brother? Didn’t he say he wanted to go to summer camp?” James gets closer to Isaiah. He puts his arms around his shoulder and says, “Don’t look so shocked Mr. Isaiah. I just want to help you by helping me.”

Isaiah pushes him off and says, “Look man, I don’t know who told you all that but this is just weird. I don’t need this and I told you, I have a job. I’ll take care of my stuff on my own. Thanks, but no thanks.”

“I pay thousand a week, off the books, only need you when I call; you could go back to school, Mr. Isaiah.”

Isaiah is startled; he doesn’t know what to say. He knows he needs the money, but what does this man want from him? Why is he so interested in him and how does he know so much about him?

“Mr. Isaiah, all I ask is for one thing. Here is my offer. You meet me here, at Belvedere Castle, when I call, and you take a package, just like that one.” James turns around and points at a cardboard box. Apparently the box has been there a while, but Isaiah had not noticed it.

“Nahhhhhhhhhh man. Look I don’t do that. I don’t transport no drugs, no organs, no people.”

“But Mr. Isaiah, I haven’t even told you what the box contains. Why are you jumping to conclusions?”

“This is all weird. You have a supposed business operating from a tourist attraction in Central Park, a park, man, and you expect me to take some god damn box from you and take it who knows where? On top of that, you know all this about me. Like, who are you? You think I’m dumb, just ‘cause I’m from the Bronx? Nah. I ain’t taking that. I may need the money but I don’t want to end up in jail.”

James nods at Isaiah and goes towards the box; he takes out a box cutter from his back pocket and opens it. He turns to Isaiah and points to the contents of the box. Isaiah is blinded by his curiosity and, without hesitation, walks towards the box and looks inside. James has a smile on his face.

“Now Mr. Isaiah, what will it be?”

### **Flowcharting Assignments:**

- 1) Isaiah has a tough life, he works long hours for little money. Write a flowchart to estimate how much money Isaiah makes per week as a dishwasher? The flowchart should also calculate how many hours of sleep he gets every week. How many hours are left per week?
- 2) Think of three things that you think could be in the box that would make Isaiah decide to take the job. Create a flowchart that asks a user for the content of the box. If what the user writes is one of the three things that Isaiah would accept, it would output the message “Isaiah is taking the job!” If the user enters anything other than the three acceptable contents of the box, the output message should be “Isaiah is not taking the job!”

### APPENDIX 3: Guidelines given to instructors to implement the case studies

The goal of implementing the three case-studies in the PS course sections is to reinforce learning of flowcharting concepts and skills using Visual Logic. The general idea is to treat these case studies as practice/reinforcement assignments that students will do in class and outside class, either in groups or individually (see below). Each case study consists of a text (a story) between 2–4 pages, and some Visual Logic assignments related to the story. To make sure that students take this seriously, you may want to give some credit for the completion of the case studies by students, either as participation, classwork or homework.

After you have covered the basics of sequencing, selection and repetition in Visual Logic, the case studies should be done in the following order:

1) *Mind Games*. (GUIDED/GROUP/IN CLASS).

- a. BEFORE CLASS: students should read the case study before coming to class. Distribute the case study to students and ask them to read the stories before they came to the class session in which you will implement the case. To make sure they read the story, you could ask them to write a short summary of the story (one paragraph, 50-100 words) and to highlight important information in the story.
- b. IN CLASS: This case study should be guided and done during class time. Students will work in groups of 3-5 students trying to solve the assignments at the end of the case study. The instructor should help students if they get stuck, not by doing the assignments but by providing hints at the solution. The instructor should make sure that, in the end, all groups get a correct answer. It should take about 1 hour.
- c. COLLECT MATERIAL: Collect individual summaries and the group responses (flowcharts) preferably in electronic format with students' names.

2) *Belvedere*. (GROUP/IN CLASS).

- a. BEFORE CLASS: students should read the case-study before coming to class. Distribute the case study to students and ask them to read the stories before they came to the class session in which you will implement the case. To make sure they read the story, you could ask them to write a short summary of the story (one paragraph, 50-100 words) and to highlight important information in the story.
- b. IN CLASS: This case study should be done in groups with the instructor providing minimal guidance. It is like a group test. The role of the instructor is to only clarify the questions without providing hints at the answers. It should be done in 1 hour.
- c. COLLECT MATERIAL: Collect individual summaries and the group responses (flowcharts) preferably in electronic format with students' names.

3) *Shell Sleuth*. (INDIVIDUAL/OUTSIDE CLASS). This case study should be done individually outside the class. You could assign it as a homework assignment. The assignment should include a summary of the story, highlight important information, and the solution of the flowcharts at the end. Collect individual summaries and flowcharts preferably in electronic format.

AFTER THE COMPLETION OF THE CASE STUDIES:

- 1) We will provide you with a brief survey to gauge students' reactions to this pedagogical model. There will also be a brief survey for instructors. The survey should be anonymous.
- 2) Incorporate the 10 COMMON QUESTIONS in your assessments, exactly as you did in Fall 2014.
- 3) Send us the performance of each student (we need the names) in each of the 10 questions, graded from 0-10 with the rubric we discussed in the August workshop. Partial credit should be given to the input, processing and output part of the solutions.