



2014 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
SCIENCE, TECHNOLOGY, ENGINEERING, MATH & EDUCATION
JUNE 16, 17, & 18 2014
ALA MOANA HOTEL, HONOLULU, HAWAII

ENCOURAGING STUDENT INTEREST IN QUANTITATIVE REASONING COURSES

YI, TONG

IOWA WESLEYAN COLLEGE
MOUNT PLEASANT.

Tong Yi
Iowa Wesleyan College
Mount Pleasant.

Encouraging Student Interest in Quantitative Reasoning Courses

The following approaches are useful in teaching programming languages and quantitative reasoning courses. The approaches include: designing projects with animation and multimedia effects to help students visualize results; using real life examples to explain intuitive ideas of mathematical concepts/formulas; giving step by step illustration; and increasing lab time.

ENCOURAGING STUDENT INTEREST IN QUANTITATIVE REASONING COURSES

INTRODUCTION

It is important to teach quantitative knowledge to students [1]. When students show interest in scientific fields, but lose interest after encountering difficulties, one must stimulate students so that they feel interested in the topics.

The following approaches are helpful in stirring up interest in students as a part of teaching practices.

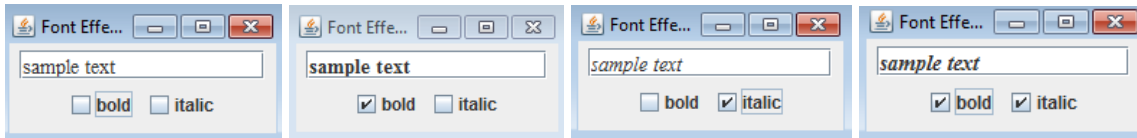
1. Design projects with animation and multimedia effects to help students visualize results. As stated in [2], brain learns better with images.
Students are not enthusiastic about exercises that solely stress number crunching. For example, a student once asked, “Why do we need to write a program to add two numbers? Why not simply use a calculator?” For many students, finding the greatest common divisor of two integers is not as interesting as drawing a snowman.
2. It would be a good idea to explain concepts and formulas with images.
Students would be frustrated to be told, “Here is the formula. Just use it.”
Use real life examples to explain the intuitive idea of mathematical concepts/formulas. Finding out how a formula is related to daily lives helps students understand it better. For example, once students apply Euclid’s algorithm to find the maximum size of square tiles that evenly cover a floor, they might remember the algorithm better.
3. Give step by step illustration of algorithms/calculations.
When teaching beginners, it is important to give detailed explanation, accompanied with figures.
4. Adopt “Learn by doing” approach:
Let students work on pair programming and then test their results in computers. Students like the idea that “I can do it and see the result immediately.”

In the following sections, find the above points illustrated with projects and examples. The author uses all of them in Programming I and II courses.

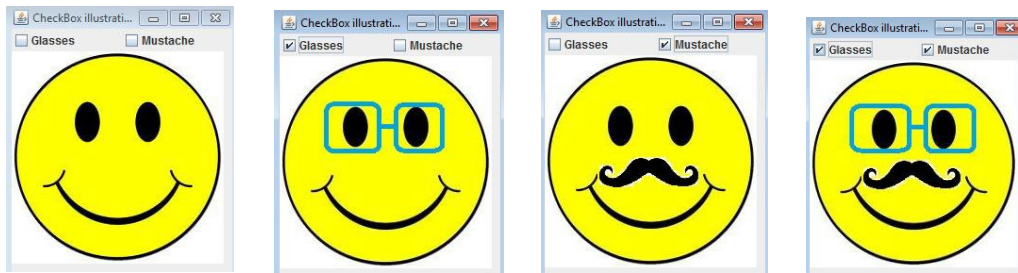
DESIGN PROJECTS THAT HELP STUDENTS TO VISUALIZE RESULTS

Checkbox example

In most current textbooks [2-8], checkboxes are shown to change font styles.



By comparison, some students prefer to use checkboxes to add/remove glasses or a mustache to/from a smiley face. It is inspired by [13].



Students were even more engaged in a slide show project with smiley face animations.

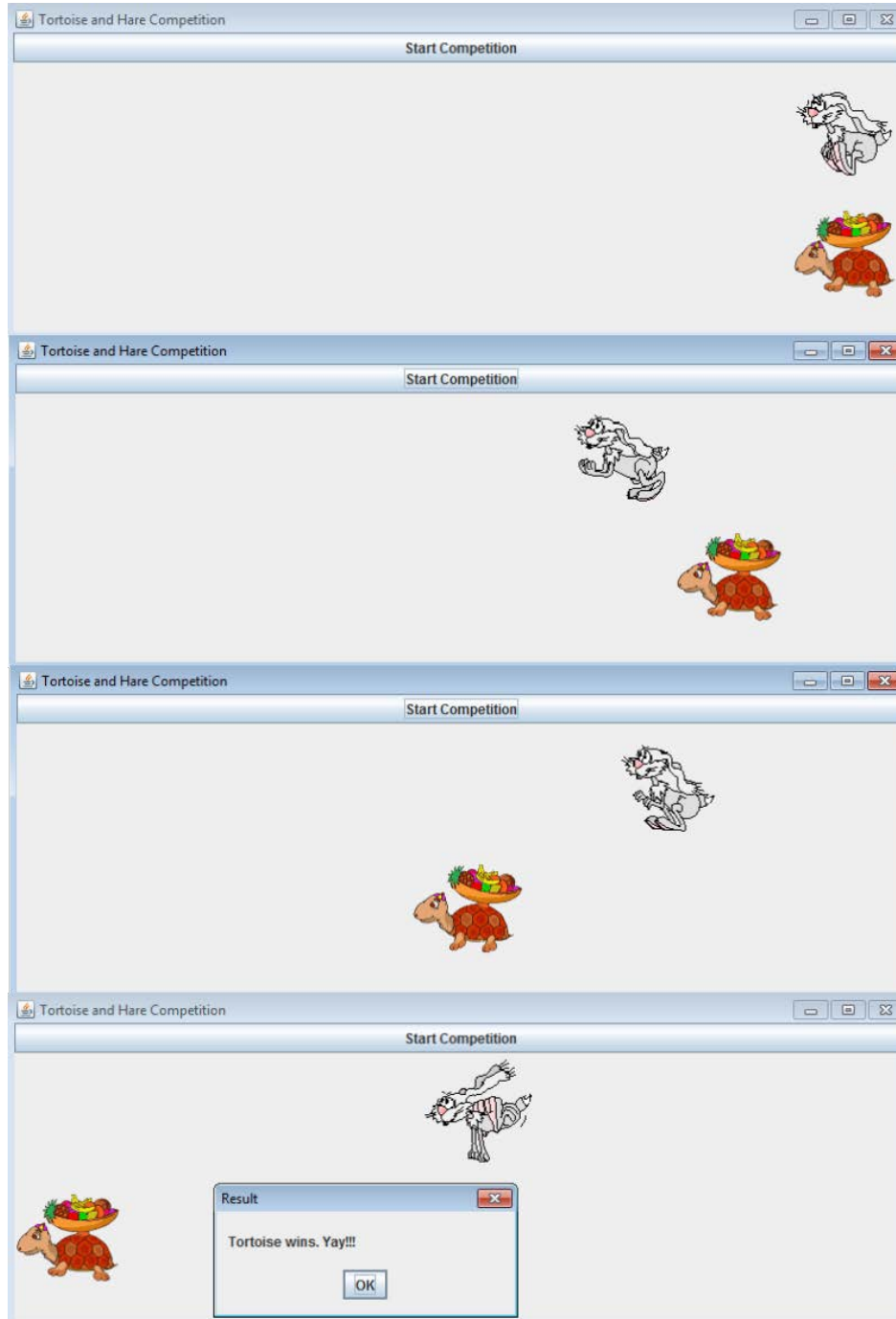
Tortoise and hare competition simulation

We are all familiar with Aesop's famous fable on the Tortoise and the Hare. The following is a sample output of simulating a competition between them, where letter 'T' means the position of Tortoise and 'H' means that of Hare [4]. If the hare and the tortoise encounter each other at the same position, only letter 'H' is shown. The road is slippery, thus the animals may fall backwards. The competition is finished if at least one of the animals reaches the end of the road.

```

Time:1      H
Time:2      H T
Time:3      H   T
Time:4          H
Time:5          H T
Time:6          H T
Time:7          T   H
Time:8          T   TH
Time:9          T   T   H
Time:10         T   T   H
Time:11         T   T   H
Time:12         T   T   H   H
Time:13         T   T   H   H
Time:14         T   T   H   H
Time:15         T   T   H   H
Time:16         T   T   H   H
Time:17         T   T   H   H
Time:18         T   T   H   H
Time:19         T   T   H   H
Time:20         T   T   H   H
Time:21         T   T   H   H
Time:22         T   T   H   H
Hare wins. Yuck.
  
```

Students like this project since it shows that there are interesting applications other than crunching numbers. However, they want to see the competition with more visually appeal, with effects close to an animated movie. The following is an updated version with animated figures, which are downloaded from [14, 15]. We also add background music to make the competition look like a real movie.

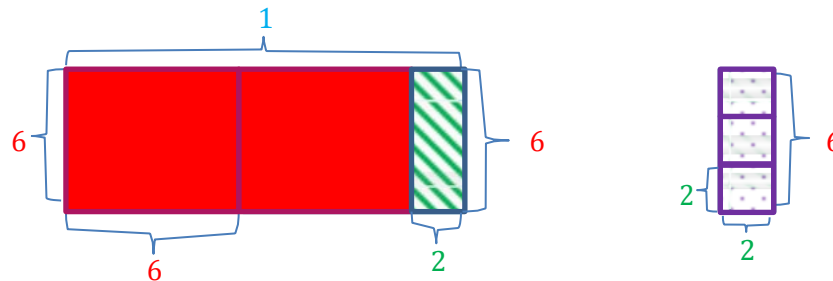


USE REAL LIFE EXAMPLE TO EXPLAIN THE INTUITIVE IDEA OF MATHEMATICAL CONCEPTS/FORMULAS

A formula can be better understood if it is related with our daily life. Here is an example on using Euclid's algorithm to find greatest common divisor.

Greatest common divisor

The greatest common divisor of integers m and n , represented by $\gcd(m, n)$, can be interpreted as the maximum size of squares that evenly cover an m by n floor. For example, the maximum size of squares that evenly cover a 14×6 floor is 2. That is, $\gcd(14, 6) = 2$. In the following, we show how to calculate $\gcd(14, 6)$.



We first try to cover a 14×6 floor with 6×6 squares. There is a 6×2 area left uncovered, where 2 is the remainder of 14 divided by 6, as shown in the above left figure. This implies that $\gcd(14, 6) < 6$. Intuitively, the maximum square tile to cover the remaining 6×2 floor coincides with the maximum square tile to cover the original 14×6 floor. Next, we shall prove that $\gcd(14, 6) = \gcd(6, 14 \% 6)$, where $\%$ is the remainder operator.

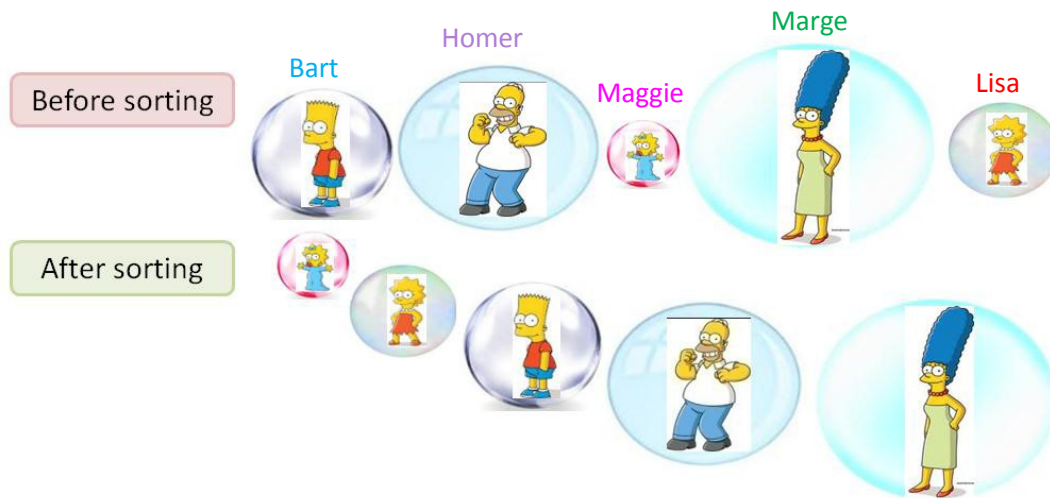
Note that any square that covers 6×2 can evenly cover 14×6 (one approach is to use those squares to cover 6×2 first. Afterwards, use them to cover 12×6). Hence, we have $\gcd(14, 6) \geq \gcd(6, 2)$.

Assume that $\gcd(14, 6) > \gcd(6, 2)$. Since $\gcd(14, 6)$ is a factor of 6, it is also a factor of 12, thus, square with size $\gcd(14, 6)$ covers 12×6 evenly. Hence $\gcd(14, 6)$ should evenly cover the remaining 6×2 . By assumption, $\gcd(14, 6)$ is larger than $\gcd(6, 2)$, this implies that $\gcd(14, 6)$ cannot evenly cover 6×2 ; a contradiction is introduced.

From the above proof, we have $\gcd(14, 6) = \gcd(6, 14 \% 6)$. In general, $\gcd(m, n) = \gcd(n, m \% n)$, as stated in Euclid's algorithm.

STEP BY STEP ILLUSTRATION OF ALGORITHMS/CALCULATIONS

It is important to give a step by step explanation when teaching beginners. We use the Simpsons to illustrate the operations of Bubble sort, where the larger number, stored in the larger bubble, sinks down to the bottom while the smaller number, stored in the smaller bubble, flows up to the front.



The following procedure shows how to sink the largest bubble to the rightmost side, thus putting the largest bubble in the correct position. Afterwards, we sink the second largest bubble to the second-to-right position, and so on. Once every bubble is in its position, we finish arranging the bubbles in ascending order.

We first define the order of an adjacent pair: an adjacent pair is in-order if the left bubble is no larger than the right bubble; otherwise, that pair is out-of-order. For example, Bart and Homer adjacent pair is in-order, and Homer and Maggie adjacent pair is out-of-order.

If every adjacent pair is in-order, then the elements are already in ascending order. Thus, our next task is to process those out-of-order adjacent pairs. Starting from the left, compare each adjacent pair and exchange if that pair is out-of-order.

- (1) Bart and Homer are in-order, so do nothing. Move to the next adjacent pair: Homer and Maggie.

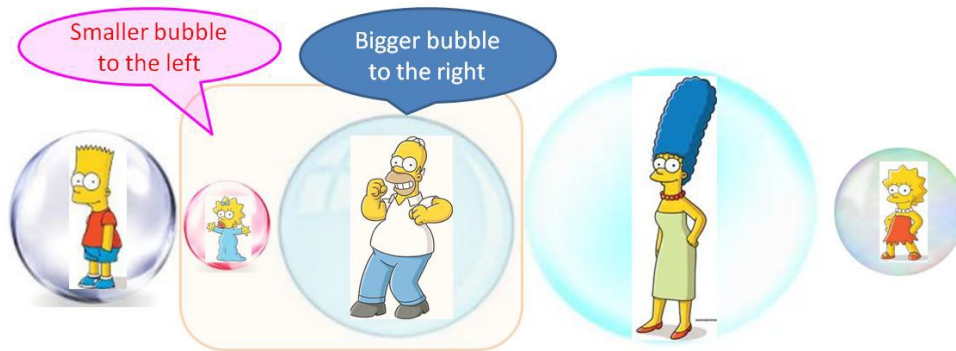


- (2) In the following figure, Homer and Maggie are out-of-order, we need to exchange



them.

After the exchange of Homer and Maggie, we get the following figure.



Move to the next adjacent pair: Homer and Marge. Due to page limit, we omit the next two steps.

- (3) After one round of comparing each adjacent pairs, Marge, the largest bubble, is in its right position.



- (4) Repeat the above steps until each bubble is in its right position.

It is a good idea to emphasize the same idea with different approaches, with the hope that at least one of them will enlighten students. Thus, besides the Simpsons' illustration, we also ask students to line up and physically act out the sorting process. An additional step is for students to watch related videos [9, 10].

LEARN BY DOING

There are ideas to be learned from “flipped classroom.”[11] However, instead of depending on students learning from videos before class and devoting most class time on exercises, lecture is still the most used delivery of need-to-know information.

It is still very important for face-to-face interaction in teaching new materials. Occasionally, the author asks students to watch videos before class. But it turns out most of them do not watch; and for the students who actually watch the videos, they do not quite understand the materials without assistance.

Most students, especially beginners, cannot learn things simply by watching videos. Questions need be addressed immediately in the learning process, or students might lose interest and will not bother to go to the next slide.

Like the “flipped classroom,” we should increase the weight of lab time compared to the traditional lectures-only approach. We let students work on paired programming [12]. Afterwards, one of them enters the program into a computer to run it. This will put

some peer pressure on students and also help them to cooperate with each other. Also, this has unexpected benefits: students like to write their code on paper before they put it to test on computer. Also, some students like to act as tutors and explain concepts to fellow students.

Finally, to stir up student interest, a musical calendar project, enhanced from [8, 16], is assigned before a new year. Each student writes code to generate a calendar that includes a recorded voice or selected songs. They can add family photos for a slide show (The illustrated picture is taken from sample pictures of Windows 7). They will send that calendar to loved ones.



CONCLUSION

This paper, illustrated with projects and notes, lists some useful approaches to teaching quantitative methods to students who may lack interest or mathematical background at the beginning of their programming studies.

The author teaches all the above materials to Programming I and II classes at a liberal art college. Students like them. The goal is to stimulate students' interests.

ACKNOWLEDGEMENT

The input from our students is much appreciated. Also, the author learned useful tricks and techniques from observing classes of colleagues. I thank P. Sanders for proofreading this paper. Finally, the author is grateful for the comments from anonymous reviewers.

REFERENCES

- [1] Carnevale, A. P., Donna M. D., "The Democratization of Mathematics." In *Quantitative Literacy: Why Numeracy Matters for Schools and Colleges*, edited by Bernard L. M., Steen L. A., Princeton, NJ: National Council on Education and the Disciplines. P21-31, 2003.
- [2] Sierra, K., Bates, B., *Head First Java, 2nd ed*, Sebastopol, CA: O'Reilly, 2005
- [3] Anderson, J., Franceschi, H., *Java 5 Illuminated*, Boston: Jones and Bartlett Publishers, 2005.
- [4] Deitel, P., Deitel, H., *Java How to Program, 8th ed*, Upper Saddle River, NJ: Pearson, 2010.
- [5] Gaddis, T., *Starting out with Java: early objects, 4th ed*, Boston: Addison-Wesley, 2011.
- [6] Horstmann, C. S., Cornell, G., *Core Java: Volume I: Fundamentals, 8th ed*, Upper Saddle River, NJ: Prentice Hall, 2008.
- [7] Lewis, J., DePasquale, P., Chase, J., *Java Foundations: Introduction to Program Design and Data Structures, 2nd ed*, Boston: Addison Wesley, 2011.
- [8] Liang, D. Y., *Introduction to Java Programming, 9th ed*, Boston: Pearson, 2013.
- [9] Mures, T., "Bubble-sort with Hungarian ("Csángó") folk dance," <http://www.youtube.com/watch?v=lyZQPjUT5B4>
- [10] Baecker, R. M., Sorting out sorting (video), <http://www.youtube.com/watch?v=SJwEwA5gOkM>
- [11] Ronchetti, M., Using video lectures to make teaching more interactive, *Proceedings of Interactive Conference on Computer Aided Learning*, 74-80, 2009.
- [12] Williams, L., Kessler, R., All I Really Need to Know about Pair Programming I Learned in Kindergarten, *Communications of the ACM*, 43(5), 108-114, 2000.
- [13] "How to Use Buttons, Check Boxes, and Radio Buttons." *The Java Tutorials*, Oracle, <http://docs.oracle.com/javase/tutorial/uiswing/components/button.html>
- [14] Hare animation picture: <http://caccioppoli.com/Animated%20gifs/Disney/The%20tortoise%20and%20the%20hare/hare.gif>
- [15] Tortoise animation picture: <http://www.bestgraph.com/gifs/animaux/tortues/tortues-12.gif>
- [16] Calendar.java code: http://www.distrodisco.com/linux_commands/lian-java-chapter-15-exercise-5/