

# Removing the Fear from Programming

Education Track  
10 June 2010

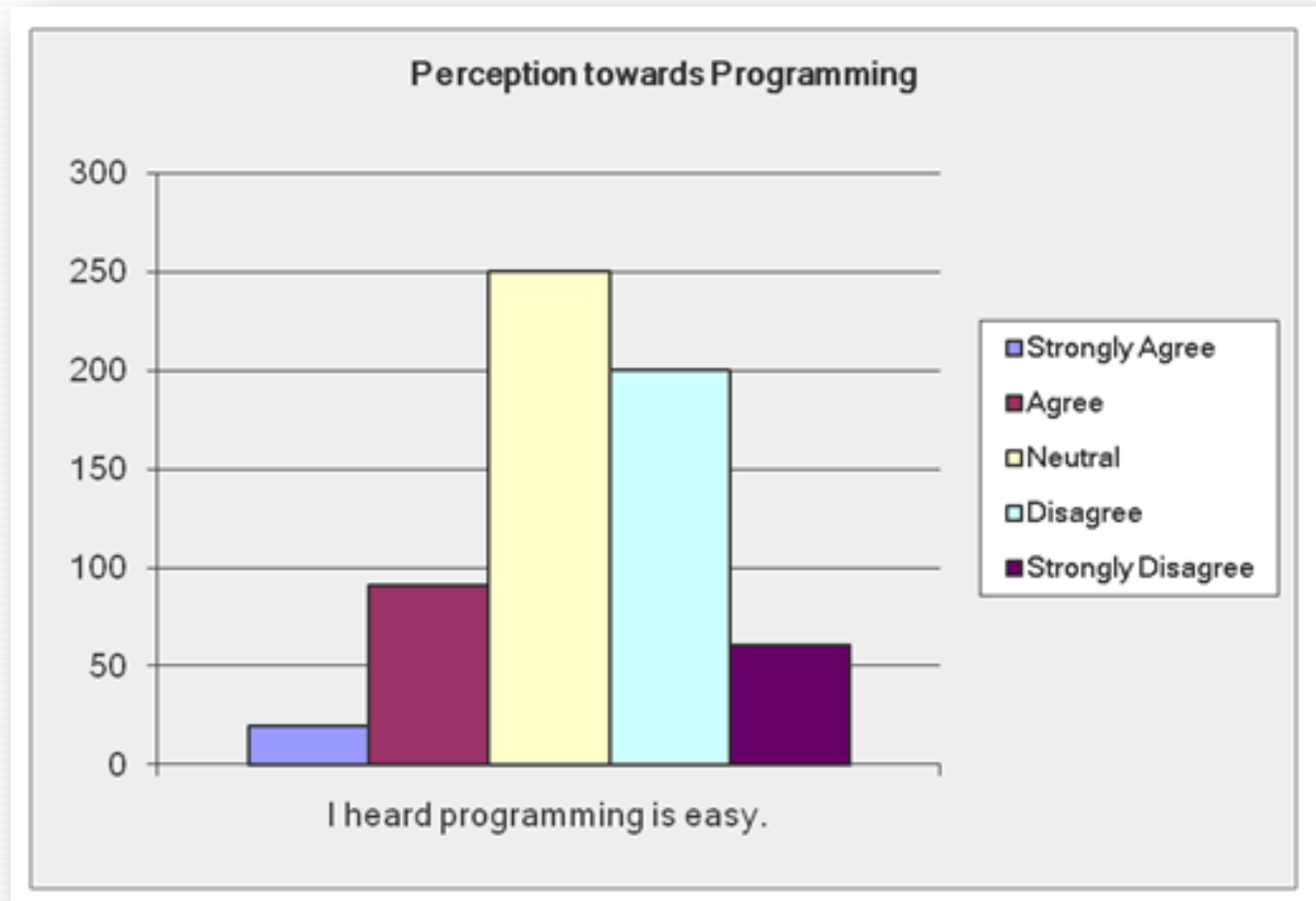
Tan Kok Cheng  
Jason Lim  
Melani Jay

# Background

- Teaching programming in Republic Polytechnic
  - Until 2008 : All year 1 students are required to go through a Mathematics and Computing Module
  - Visual Basic is the language of choice
  - In general, most students have fairly negative experience about having to learn programming
- Change Introduced in AY 2009
  - Only Engineering and IT students are required to go through the programming modules (> 2000 students)
  - A new programming module is introduced (No more mathematics)
  - **Python** is chosen as the language of choice

# Student's Perception about Programming

- The survey saw 600+ responses from the students on general perception towards learning programming
- Most students find learning programming a difficult subject



## Some feedback from students (2008 Batch after 1 module of VB programming)

- "Year 1 programming modules were in fact a **living nightmare** for me. Imagine that you had to spend long painful hours doing coding with your teammates where everyone is literally **banging their heads on the wall** while trying to get the program working just the way the problem statement wanted it to be."
- "But the experience with VB from year 1, i felt that it was challenging to learn about programming."
- "To me, the thought of learning programming really **intimidate** me. This is because I am really sucks with my programming especially VB and it really take time for me to learn something new especially programming."

# Current Situation: Problems Faced

- High Attrition rates
  - Not dropping enrollment, but dropping attendance
- Poor motivation
  - Many students are not from IT or Engineering related diploma
  - Do not see the need/relevance of learning programming
- Lack of Interest
  - Not taking a course by choice

# Possible Reasons for Bad Experience with Programming

- Choice of Programming Languages
  - Using VB (Complexity of Visual Studio)
  - Clicking buttons, drag and drop → Too much magic
- Other Factors
  - Mathematics and Computing in a single module (No continuity)
    - Students who do not like maths will start to hate programming
  - Should programming be a common Year 1 module?

# Objectives of the New Programming Module

## (Introduced in 2009)

- Equip students with Problem Solving Skills
  - Analyzing and breaking down a problem
- Cover basic programming concept/skill such as
  - Decision Making
  - Repetition
  - Modular Design
  - etc
- Address the problems highlighted
  - Dropping attendance
  - Motivation, Interest Levels
- Pick up transferable skill in learning new programming languages in higher year of studies.

# How We Deal with the Problem

- Choose a new programming language
  - VB is good for professionals for rapid prototyping
- Why we choose Python
  - Get simple things done quickly
  - Rich set of libraries (csv, PIL, pygame, etc)
  - etc

```
C  
#include<stdio.h>  
int main(int argc, char ** argv)  
{  
    printf("Hello, World!\n");  
}
```

```
Java  
public class Hello  
{  
    public static void main(String argv[])  
    {  
        System.out.println("Hello, World!");  
    }  
}
```

```
Python  
print "Hello, World!"
```

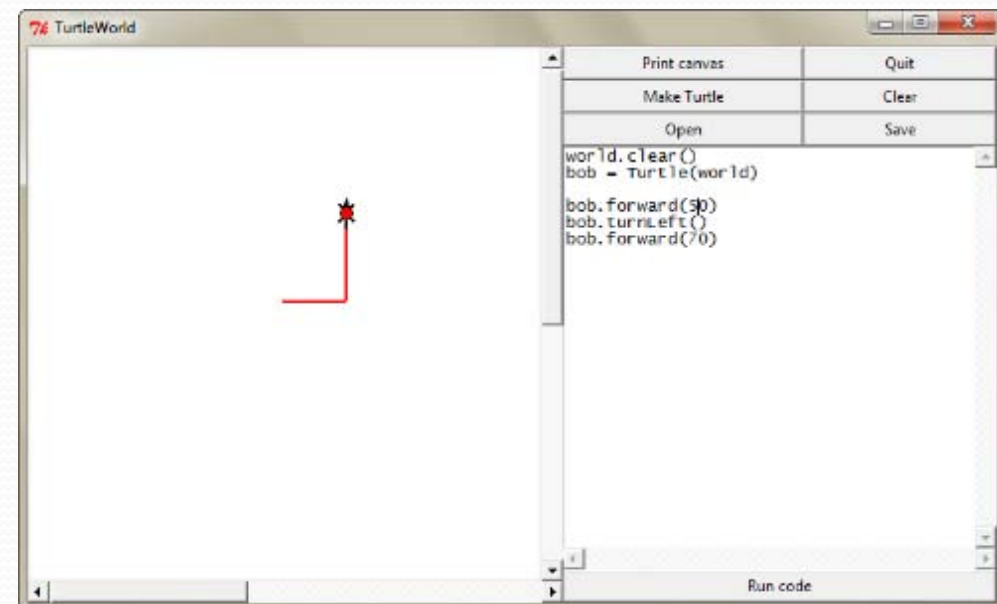


# How We Deal with the Problem (II)

- Curriculum Design
  - Focus on Graphical, Visual and Interactive problem
  - Create Games and other (Tangible) Things
  - Focus more on Problem Solving and less on syntax
- Targeting the “right” students
  - Not a common Year 1 Module Anymore
  - Only for Engineering and IT students

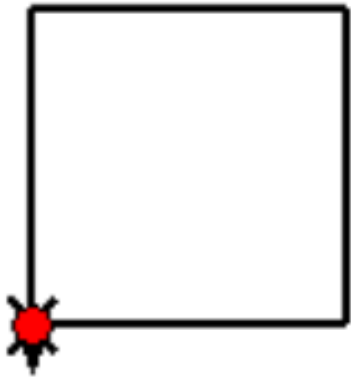
# Case Study: The TurtleWorld

- An environment to learn programming by instructing a turtle to draw picture
- A turtle has 3 attributes:
  - Position, Orientation, Pen Up or Down
- The programmer should understand (and predict and reason about) the turtle's motion by imagining how they would move if they were the turtle



# Turtle Graphics: Drawing Simple Shapes

- A combination of commands enables the turtle to perform a series of actions
- Example: to instruct the turtle to draw a square

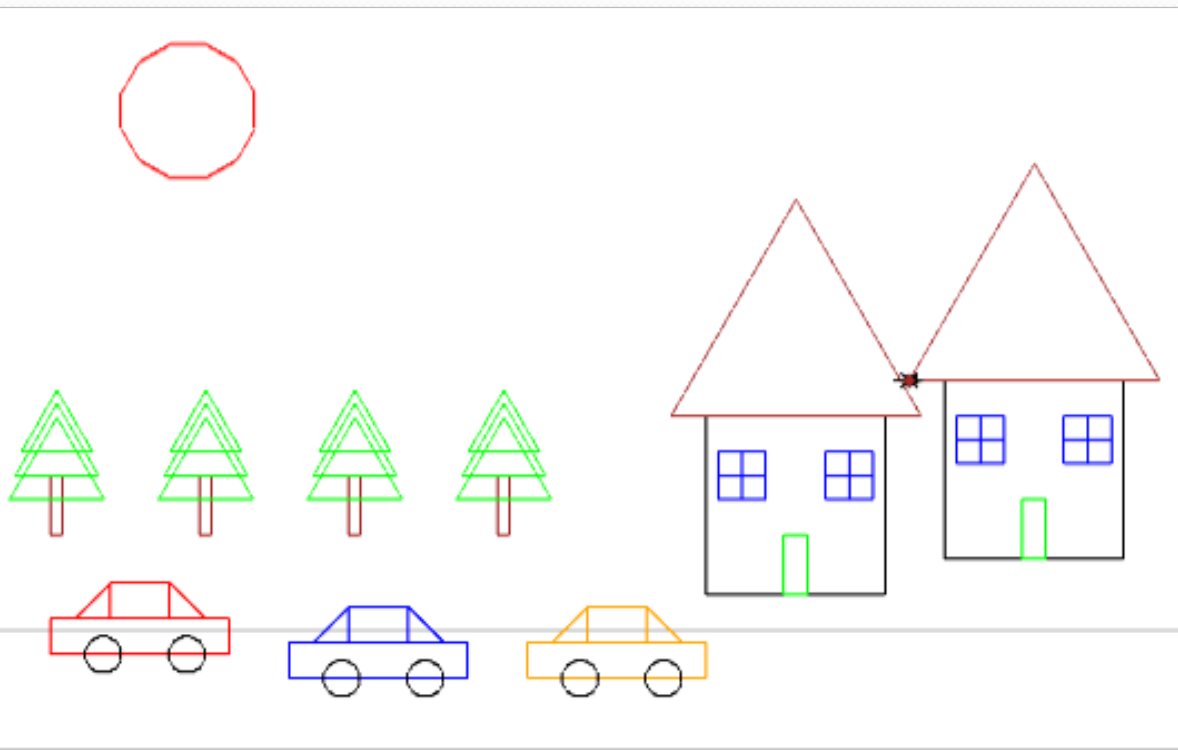


```
1 bob.forward(80)
2 bob.turnLeft()
3 bob.forward(80)
4 bob.turnLeft()
5 bob.forward(80)
6 bob.turnLeft()
7 bob.forward(80)
```

*Every command is executed in sequence, with each command resulting in a single action*

# Turtle Graphics: Create Generic Functions

- Analyze the drawing shown below:
  - Drawing is created with a collection of simple shapes
  - Shapes used: Triangle, Circle, Rectangle, Square
  - Each shape has different color, size and orientation

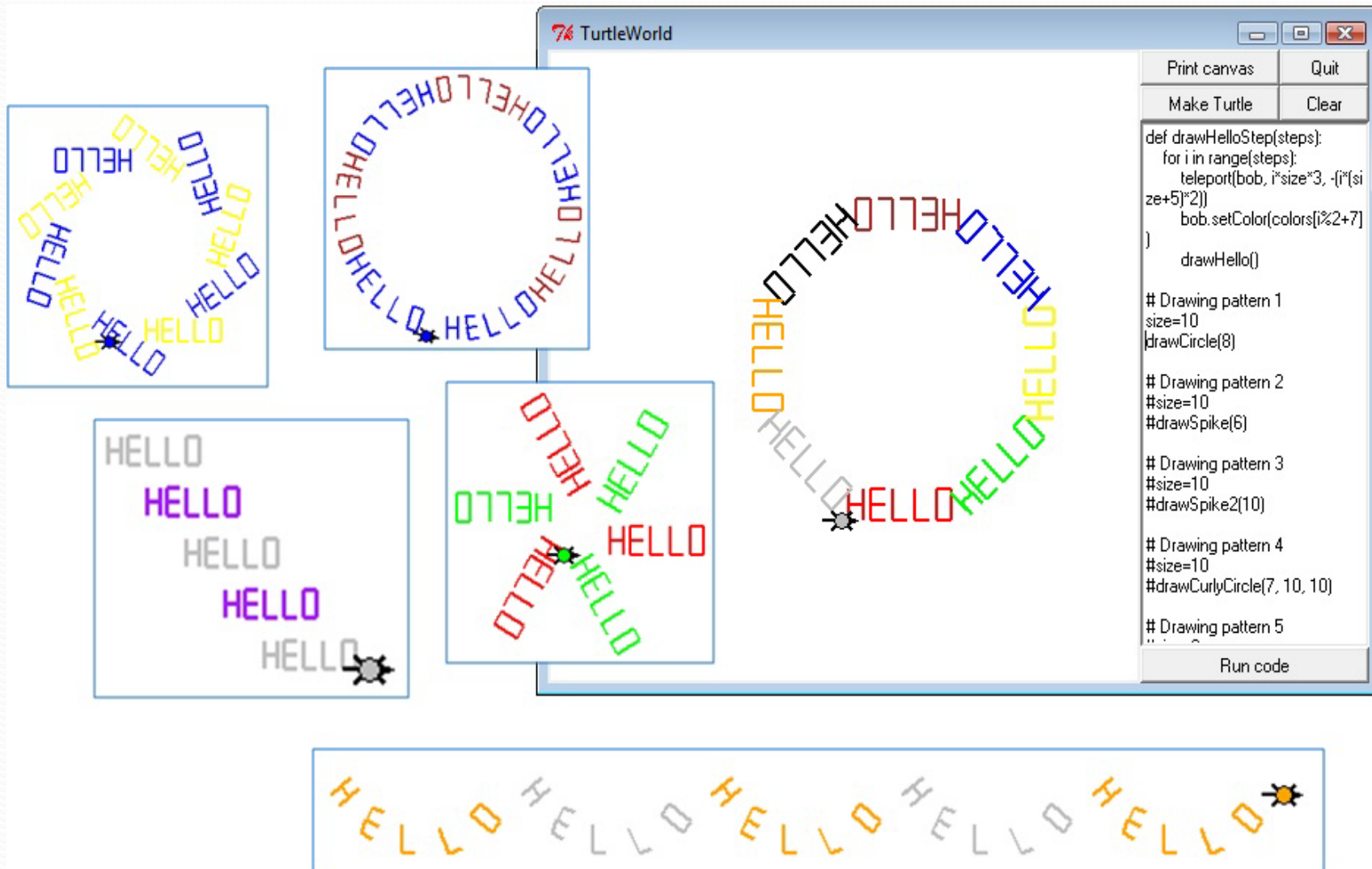


1. Define a function for each shape.

2. Each function must be able to draw shape at different size, color and orientation

# Turtle Graphics: Repetition

- Give instructions to guide the turtle to generate a series of Word Art.



The image displays the TurtleWorld software interface, which includes a canvas and a code editor. The canvas shows several examples of word art generated by a turtle. The code editor contains the following Python code:

```
def drawHelloStep(steps):  
    for i in range(steps):  
        teleport(bob, i*size*3, -(i*(size+5)*2))  
        bob.setColor(colors[(i%2+7)])  
    drawHello()  
  
# Drawing pattern 1  
size=10  
drawCircle(8)  
  
# Drawing pattern 2  
#size=10  
#drawSpike(6)  
  
# Drawing pattern 3  
#size=10  
#drawSpike2(10)  
  
# Drawing pattern 4  
#size=10  
#drawCurlyCircle(7, 10, 10)  
  
# Drawing pattern 5  
..
```

The code editor also features buttons for "Print canvas", "Quit", "Make Turtle", and "Clear", and a "Run code" button at the bottom.

The word art examples include:

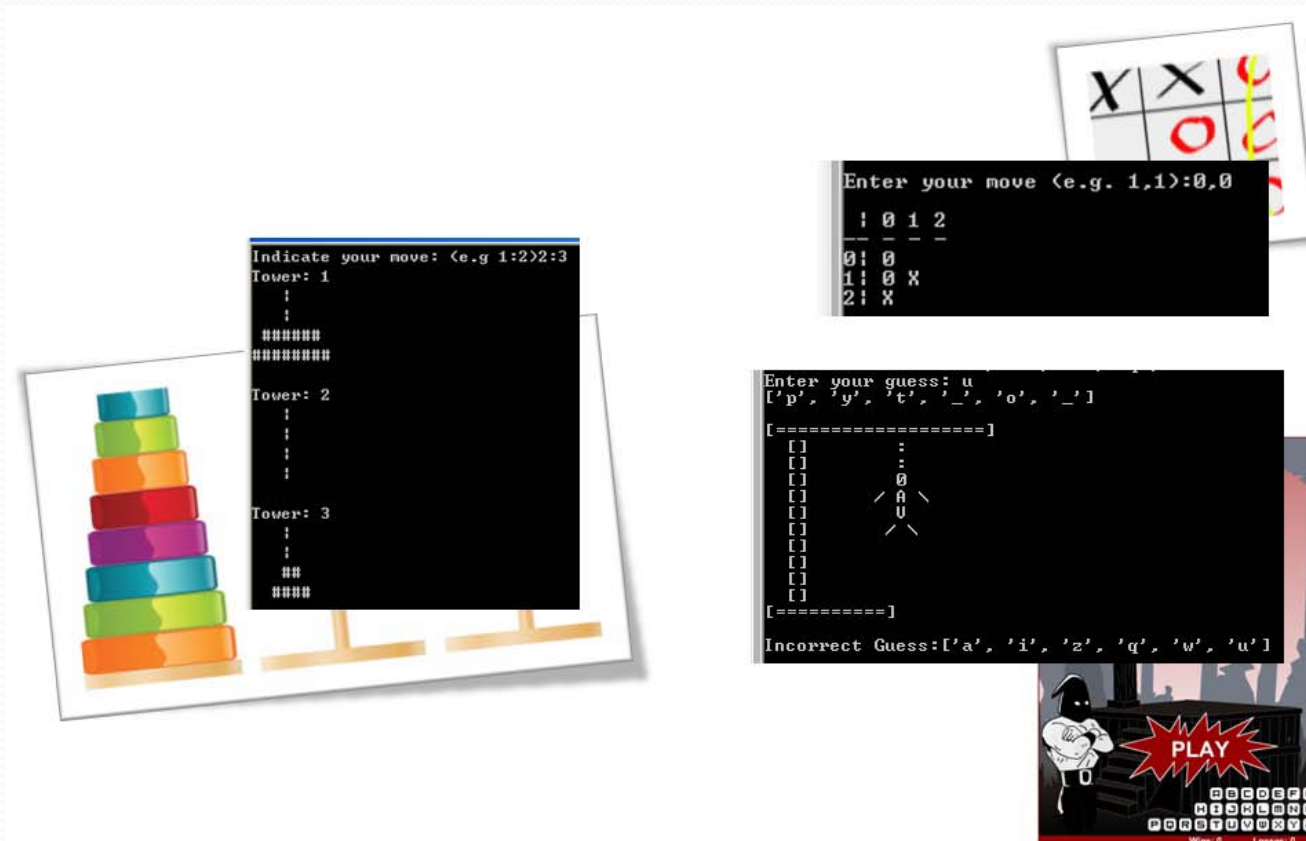
- A circular pattern of "HELLO" words in blue and yellow.
- A circular pattern of "HELLO" words in red and blue.
- A circular pattern of "HELLO" words in red, blue, and green.
- A vertical stack of "HELLO" words in purple and grey.
- A pattern of "HELLO" words in red and green, arranged in a cross-like shape.
- A horizontal row of "HELLO" words in orange and grey.

# Programming Concepts Covered in Turtle Graphics

- Modular Design: Using Functions
- Generic/Flexibility: Using Generic Functions
- Repetitions: Using While and for Loops

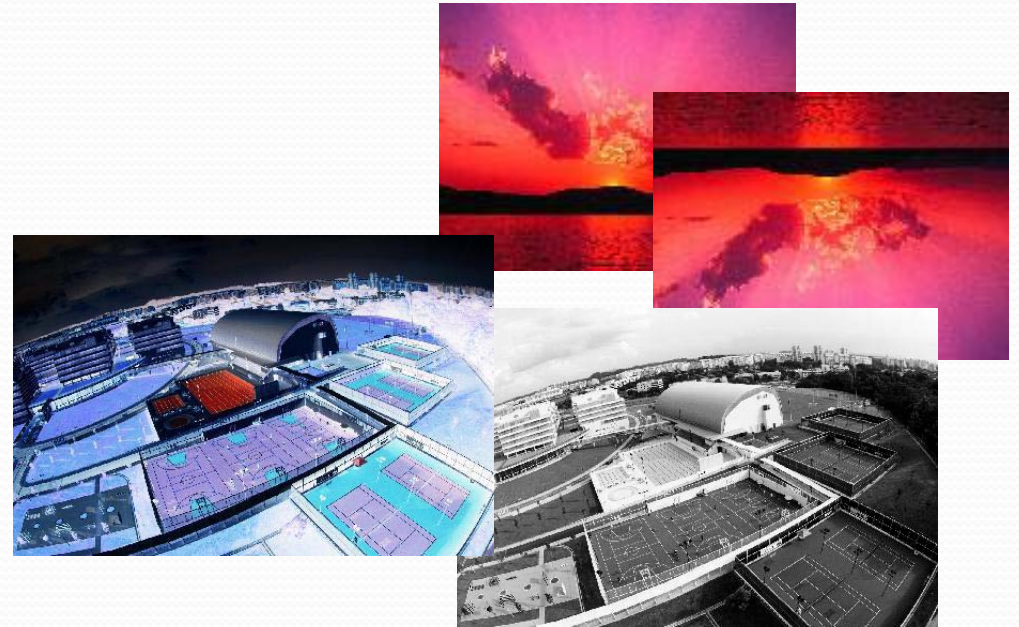
# Create Games

- Bring up the fun in programming (hopefully)
  - Simplicity of getting user input: `input()/ raw_input()` enables the creation of simple console games



# Create (Tangible) Things

- Image Processing with PIL
  - Resizing
  - Transforming
  - Filtering





# Problem Solving vs Learning Syntax

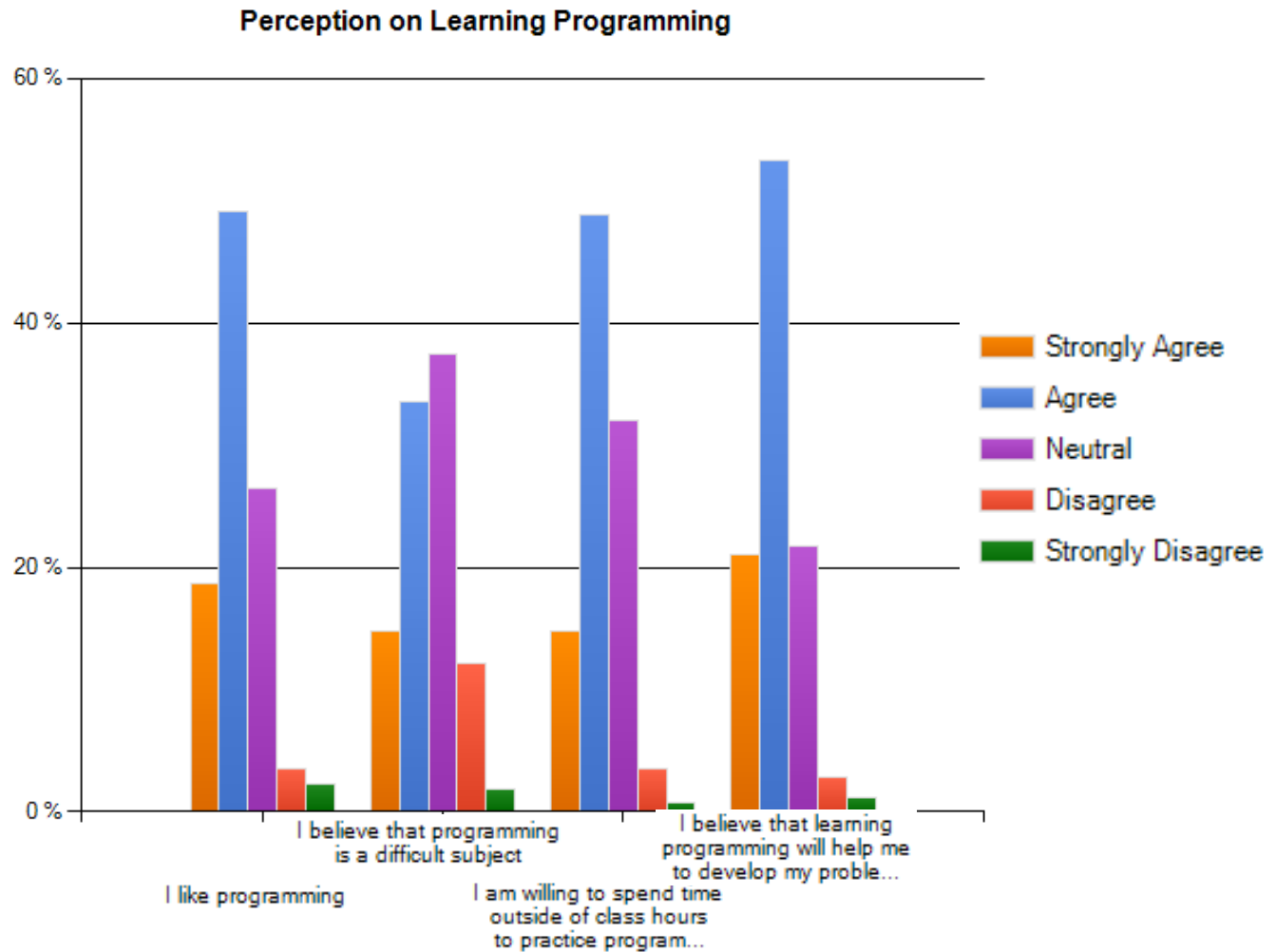
- Inline with RP's Problem based Learning Approach
- Use problem trigger to bring out the relevance of learning programming
  - Create Drawing (in Turtle Graphics)
  - Create Games
  - Create Things

# Results/ Outcomes

- Survey
- Attendance
- Student's feedback

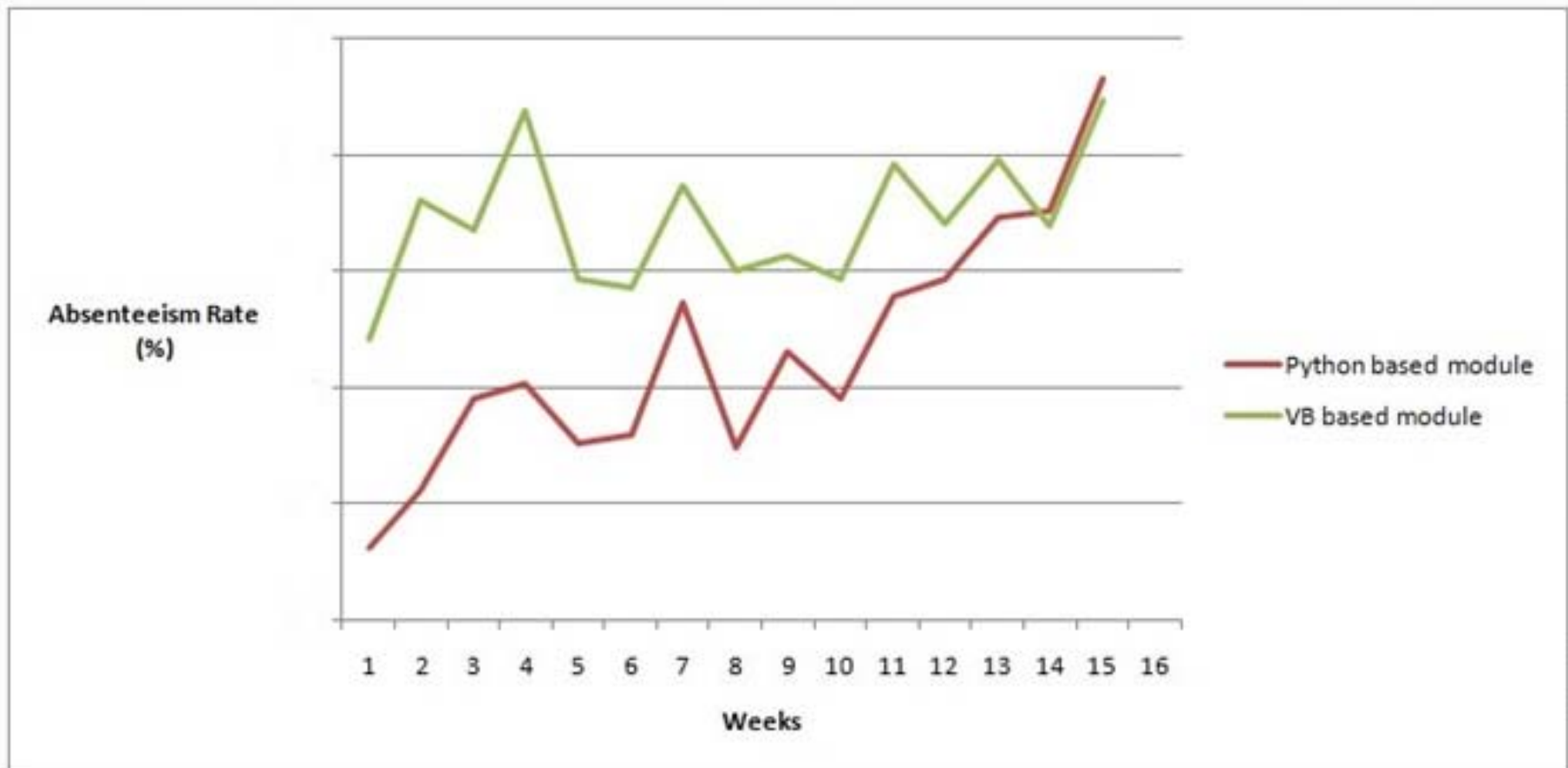
# Students' Perception on Programming (After 4 lessons)

- The survey saw 600+ responses from the students on general perception towards learning programming
- 4 categories were being asked
  - I like programming
  - I believe that programming is a difficult subject
  - I am willing to spend time outside of class hours to practice programming
  - I believe that learning programming will help me to develop my problem solving skills



# Absenteeism Rate

- Comparing attendance over 15 lessons
  - VB/Maths module (2008)
  - Python module (2009)



# Students' Feedback

## **Lots of opportunities to expand ideas**

I am very interested in Turtle World to create different shapes and colors. Also the python software is very nice to create different programs according to our wish. In one earlier class facilitator gave us the work to design a program in python for selecting your ideal life partner. It was fun. Also in future through programming I can study to create computer games and more.

## **A sense of satisfaction:**

Learning programme is interesting as there are a lot of new things to learn and new moves to input. ***the most enjoyable part is the moment when you see your programme run as that is your creation. so amazing and interesting.***

Sometimes it may be frustrating but when i complete my project, I feel very satisfied. Because I know that my hardwork results in my good work done

Interesting and funny! ***Feel proud when i made a successful program.*** :) I think programming can help me on work somehow.

I think that programming is fun! I enjoyed the lesson alot. As it maybe tough at first but after doing it and ***seeing the result make me to have satisfaction.***

# Students' Feedback

## Change of perception:

When first time I heard programming, I thought that is a hard for me to learn. But after I learnt, that is not very hard to learnt. It is very east to programming the turtle that we have learnt in the first time in class

Well it seem scary at first but as time goes by it appeared easy and things started to flow smother which ain't an issue to me.

***At first, i heard loads of seniors saying programming is very boring but after attending the lesson, i realise it was not as boring as i thought and i'm amazed by programming can do***

# Conclusion

- Most students have positive feedback about Turtle Graphics
- There is some improvement in attendance
- More students can find the relevance of learning programming. (More positive feedback received)
- Students doing Java in Year 2 are able to relate back to what they have learnt in Python in Year 1.