

**DSM Academy**  
**Computer Science Courses (2021 Fall)**

All courses will be delivered via Zoom. Lecture recording and notes will be available to students.

**\*Early bird discount:** sign up and pay tuition by **August 22, 2021**.

**Course Registration:** <https://www.dsm-academy.net/>

**Contact:** [DSMAcademy.Shen@gmail.com](mailto:DSMAcademy.Shen@gmail.com), [js48@txstate.edu](mailto:js48@txstate.edu)



**WeChat:** jianshentx

**DSM Academy WeChat Group:**



15 weeks (August 27 – December 11, 2021)

Meet once a week on Friday or Saturday

Total class meetings: 15

No class on Nov. 26, 27 (Thanksgiving)

Course/ Instructor	Time	Prerequisite	Tuition
Scratch Programming Dr. Wenbin Luo	6:30—7:30 PM (Central Time) Friday	Grades 3--8	\$300 (regular) Early Bird: \$280
Python Programming Dr. Wenbin Luo	5:00—6:15 PM (Central Time) Friday	Grades 5--11	\$300 (regular) Early Bird: \$280
Java Programming Dr. Wenbin Luo	6:00—7:15 PM (Central Time) Saturday	Grades 7—12	\$300 (regular) Early Bird: \$280

### **Scratch Programming**

Scratch is a block-based visual programming language, designed primarily for young students as an educational tool for coding. This course will teach students the computational thinking skills highly sought-after in our digital world. Students will have the opportunity to learn programming through hands-on projects, such as drawing fractal trees, simulating rocket launching, fireworks & snow simulation, building an analog & digital clock, bubble sort, linear & binary search, and encryption & decryption etc. Students will be able to showcase and share their programming products online with family and friends. Scratch programming is ideal for students 8 to 16 years old.

### **Python Programming**

Python is one of the most popular programming languages among data scientists and machine learning researchers. It is a perfect language for students to learn fundamental programming

skills and concepts. In this course, students will not only build solid programming skills, but also master Python specific features. The instructor will guide students to solve some programming competition problems using Python 3, which United States of America Computing Olympiad (USACO) officially supports. Python programming is ideal for students 11 to 19 years old.

### **Java Programming**

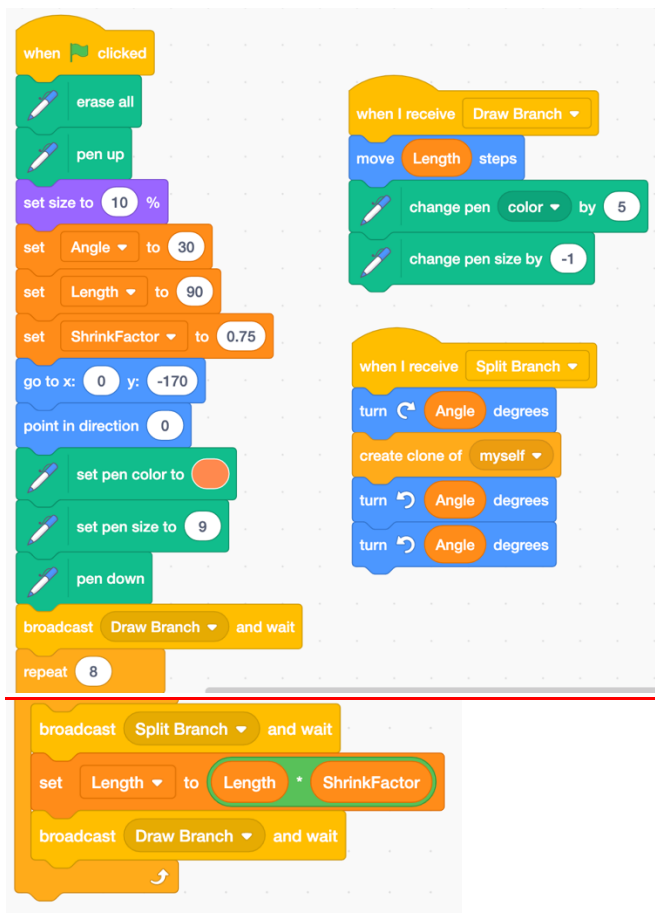
Java is another popular programming language. According to Oracle, Java is the #1 developer platform in the world with more than 10 million Java developers and 13 billion devices run Java. In this course, students will learn fundamental programming and computational thinking skills. In addition, students will master Java specific features. Java is one of the three programming languages (Python, Java, and C++) that USACO officially supports. The instructor will guide students to solve some programming competition problems from USACO and Canada Computing Olympiad etc. Java programming is ideal for students 14 to 19 years old.

### **About the Instructor: Dr. Wenbin Luo**

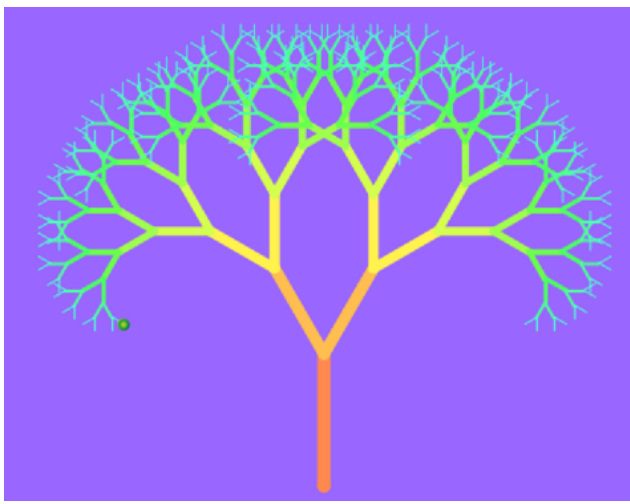
- Professor of Computer Engineering, who enjoys teaching programming (*Scratch, Python, Java, C/C++, Processing, Racket, ReasonML, Scala, and Julia etc.*)
- Ph.D. in Computer Engineering
- Oracle Certified Professional, Java SE 8 Programmer
- Senior Level Linux Professional (LPIC-3)
- More than 15 years of experience in teaching summer camps to K-12 students.

To help students and parents decide which course better fits a student's level, we included a sample program written in Scratch, Python, and Java, respectively, to draw a fractal tree, as shown below.

## A Scratch program which draws a fractal tree



## The fractal tree drawn by the Scratch program above



### A Python program which draws a fractal tree

```
import pygame, math

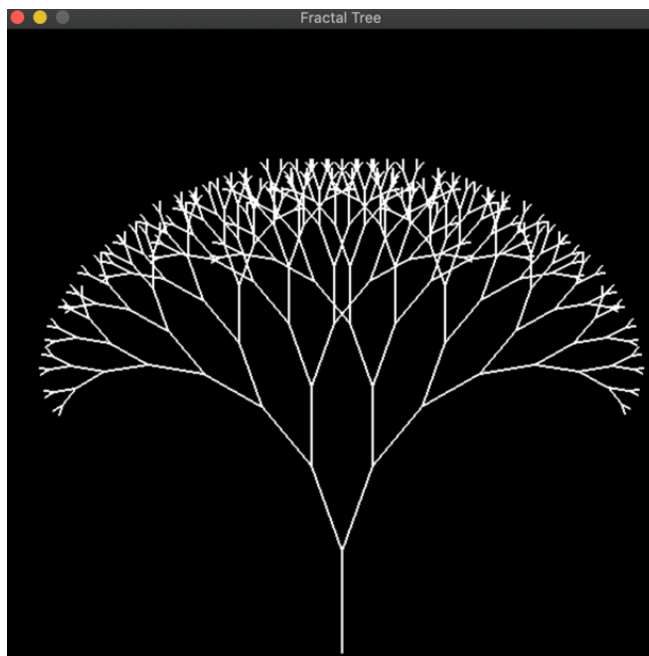
pygame.init()
window = pygame.display.set_mode((600, 600))
pygame.display.set_caption("Fractal Tree")
screen = pygame.display.get_surface()

def drawTree(x1, y1, angle, depth):
    fork_angle = 20
    base_len = 10.0
    if depth > 0:
        x2 = x1 + int(math.cos(math.radians(angle)) * depth * base_len)
        y2 = y1 + int(math.sin(math.radians(angle)) * depth * base_len)
        pygame.draw.line(screen, (255,255,255), (x1, y1), (x2, y2), 2)
        drawTree(x2, y2, angle - fork_angle, depth - 1)
        drawTree(x2, y2, angle + fork_angle, depth - 1)

def input(event):
    if event.type == pygame.QUIT:
        exit(0)

drawTree(300, 550, -90, 9)
pygame.display.flip()
while True:
    input(pygame.event.wait())
```

### The fractal tree drawn by the Python program above



### A Java program which draws a fractal tree

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;

public class FractalTree extends JFrame {

    public FractalTree() {
        super("Fractal Tree");
        setBounds(100, 100, 800, 600);
        setResizable(false);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    private void drawTree(Graphics g, int x1, int y1, double angle, int
depth) {
        if (depth == 0) return;
        int x2 = x1 + (int) (Math.cos(Math.toRadians(angle)) * depth * 10.0);
        int y2 = y1 + (int) (Math.sin(Math.toRadians(angle)) * depth * 10.0);
        g.drawLine(x1, y1, x2, y2);
        drawTree(g, x2, y2, angle - 20, depth - 1);
        drawTree(g, x2, y2, angle + 20, depth - 1);
    }

    @Override
    public void paint(Graphics g) {
        g.setColor(Color.BLACK);
        drawTree(g, 400, 500, -90, 9);
    }

    public static void main(String[] args) {
        new FractalTree().setVisible(true);
    }
}
```

### The fractal tree drawn by the Java program above

