Heaven Yoon

MIS 503 – Section 1

May 1st, 2026

**Programming Assignment #4**

**Source Code:**

#This program simulates the price of a stock over time and highlights its high and low point as well as its closing price

#Created by Heaven Yoon

#This imports Drawing Panel and random

```
from DrawingPanel import *
import random

 #variable to set starting stock price at 175
starting_stock_price = 175

def draw_grid_frame(panel, left, top, right, bottom):

    grid_square = 50

    # Chart background, border, grid, and labels.
    w = right - left
    h = bottom - top

    # Background
```

```
    panel.set_color("black")
    panel.fill_rect(left, top, w, h)

    # Grid lines
    panel.set_color("lightgray")
    x = left

    while x <= right:
        panel.draw_line(x, top, x, bottom)
        x += grid_square

    y = top
    while y <= bottom:
        panel.draw_line(left, y, right, y)
        y += grid_square



 # Labels the high point and low point
def draw_marker(panel, x, y, label):
    marker_size = 10
    panel.fill_oval(x - marker_size // 2, y - marker_size // 2, marker_size, marker_size, "cyan",
outline="black")

    label_x = x + 4
    label_y = y - 4

    panel.set_color("white")
```

```
    panel.font= "Times 12 bold"
    panel.draw_string(label, label_x + 4, label_y + 2)

 #Closing Price Label
def draw_close_label(panel, final_price, left, right, bottom):
    dollars=int(final_price)
    text = "Close $" + str(dollars) + ".00"
    label_w = 110
    label_x = (left + right) // 2 - label_w // 2
    label_y = bottom + 10

    panel.set_color("black")
    panel.font = "Times 12 bold"
    panel.draw_string(text, label_x + 4, label_y + 2)

#Animate the line to simulate the "price of the stock over time"
def animate_line(panel, left, top, right, bottom):
    steps = 24
    x = left
    y = (top + bottom) // 2
    dx = (right - left) // steps
    dy = dx
    price = starting_stock_price
    price_delta = 10.00
    highest_x = x
    highest_y = y
    lowest_x = x
```

```
lowest_y = y

step= 0
while step < steps:
    step += 1
    direction = random.choice([-1, 1])
    next_y = y + direction * dy
    if next_y < top or next_y > bottom:
        direction = -direction
        next_y = y + direction * dy

    if direction < 0:
        price += price_delta
    else:
        price -= price_delta

    draw_price(panel, price, right)

    if direction < 0:
        panel.set_color("green")
    else:
        panel.set_color("red")
    panel.draw_line(x, y, x + dx, next_y)
    panel.sleep(100)

    x += dx
    y = next_y
```

```
        if y < highest_y:
            highest_x, highest_y = x, y
        if y > lowest_y:
            lowest_x, lowest_y = x, y
    return(highest_x, highest_y), (lowest_x, lowest_y), price

# Creates the box to the top right to show price as it animates the line
def draw_price(panel, price, right):
    dollars = int(price)
    text= "Price $" + str(dollars) + ".00"
    label_w = 160
    label_h = 22
    label_x = right - label_w
    label_y = 45

    panel.fill_rect(label_x, label_y, label_w, label_h, "black", outline="black")
    panel.set_color("white")
    panel.font= "Times 12 bold"
    panel.draw_string (text, label_x + 4, label_y + 2)

    triangle_x = label_x + label_w - 18
    triangle_y = label_y + label_h // 2
    if price > starting_stock_price:
        triangle_color = "green"
        y1 = triangle_y + 5
        y2 = triangle_y + 5
```

```
        y3 = triangle_y - 5
    else:
        triangle_color = "red"
        y1 = triangle_y - 5
        y2 = triangle_y - 5
        y3 = triangle_y + 5

    if price !=starting_stock_price:
        panel.fill_polygon(triangle_x, y1, triangle_x + 12, y2, triangle_x + 6, y3,
fill=triangle_color)

def main():

    # Settings
    width = 750
    height = 500

    # Chart bounds
    left = 80
    top = 80
    right = width - 70
    bottom = height - 70
    panel = DrawingPanel(width, height, "light grey")

    # Title
    panel.set_color("black")
    panel.font = "Times 20 bold"
```

```
    panel.draw_string("Live Stock Price Chart", 150, 30)


    #Label Time Axis
    panel.font = "Times 14 bold"
    panel.draw_string("Time", right - 80, bottom + 10)


    arrow_x = right - 20
    arrow_y = bottom + 20
    panel.set_color("red")
    panel.fill_polygon(arrow_x, arrow_y - 3, arrow_x, arrow_y +3, arrow_x + 8, arrow_y,
fill="red")
    panel.draw_line(arrow_x - 12, arrow_y, arrow_x, arrow_y)


    #Label Price Axis
    panel.font = "Times 14 bold"
    panel.draw_string("Price", left - 50, top)
    arrow_x = left - 65
    arrow_y = top + 8
    panel.set_color("red")
    panel.fill_polygon(arrow_x - 3, arrow_y, arrow_x + 3, arrow_y, arrow_x, arrow_y - 6,
fill="red")
    panel.draw_line(arrow_x, arrow_y + 8, arrow_x, arrow_y)



    #Sets the initial stock price to $175
    starting_price_text = "$" + str(starting_stock_price)
    starting_y = (top + bottom) // 2 - 10
    panel.font = "Times 12 bold"
```

```
    panel.set_color("black")
    panel.draw_string(starting_price_text, left - 40, starting_y)



    # Draw grid
    draw_grid_frame(panel, left, top, right, bottom)


    #calls upon the functions to Simulates the "price of the stock over time", lowest price, highest price, and closing price
    high_point, low_point, final_price = animate_line(panel, left, top, right, bottom)
    draw_marker(panel, high_point[0], high_point[1], "High")
    draw_marker(panel, low_point[0], low_point[1], "Low")
    draw_close_label(panel, final_price, left, right, bottom)


main()
```

## Code Input:

Programming Assignment #4.py - C:\Users\heave\Desktop\Coding\Programming Assignment #4.py (3.14.3)

File Edit Format Run Options Window Help

```
#This program simulates the price of a stock over time and highlights its high and low point as well as its closing price

#Created by Heaven Yoon


#This imports Drawing Panel and random

from DrawingPanel import *
import random

 #variable to set starting stock price at 175
starting_stock_price = 175

def draw_grid_frame(panel, left, top, right, bottom):

    grid_square = 50

    # Chart background, border, grid, and labels.
    w = right - left
    h = bottom - top

    # Background
    panel.set_color("black")
    panel.fill_rect(left, top, w, h)

    # Grid lines
    panel.set_color("lightgray")
    x = left

    while x <= right:
        panel.draw_line(x, top, x, bottom)
        x += grid_square

    y = top
    while y <= bottom:
        panel.draw_line(left, y, right, y)
        y += grid_square


 # Labels the high point and low point
def draw_marker(panel, x, y, label):
    marker_size = 10
    panel.fill_oval(x - marker_size // 2, y - marker_size // 2, marker_size, marker_size, "cyan", outline="black")

    label_x = x + 4
    label_y = y - 4

    panel.set_color("white")
    panel.font= "Times 12 bold"
    panel.draw_string(label, label_x + 4, label_y + 2)

 #Closing Price Label
def draw_close_label(panel, final_price, left, right, bottom):
    dollars=int(final_price)
    text = "Close $" + str(dollars) + ".00"
    label_w = 110
    label_x = (left + right) // 2 - label_w // 2
    label_y = bottom + 10

    panel.set_color("black")
    panel.font = "Times 12 bold"
    panel.draw_string(text, label_x + 4, label_y + 2)

#Animate the line to simulate the "price of the stock over time"
def animate_line(panel, left, top, right, bottom):
    steps = 24
    x = left
    y = (top + bottom) // 2
    dx = (right - left) // steps
    dy = dx
    price = starting_stock_price
    price_delta = 10.00
    highest_x = x
    highest_y = y
    lowest_x = x
    lowest_y = y

    step= 0
    while step < steps:
        step += 1
        direction = random.choice([-1, 1])
        next_y = y + direction * dy
```

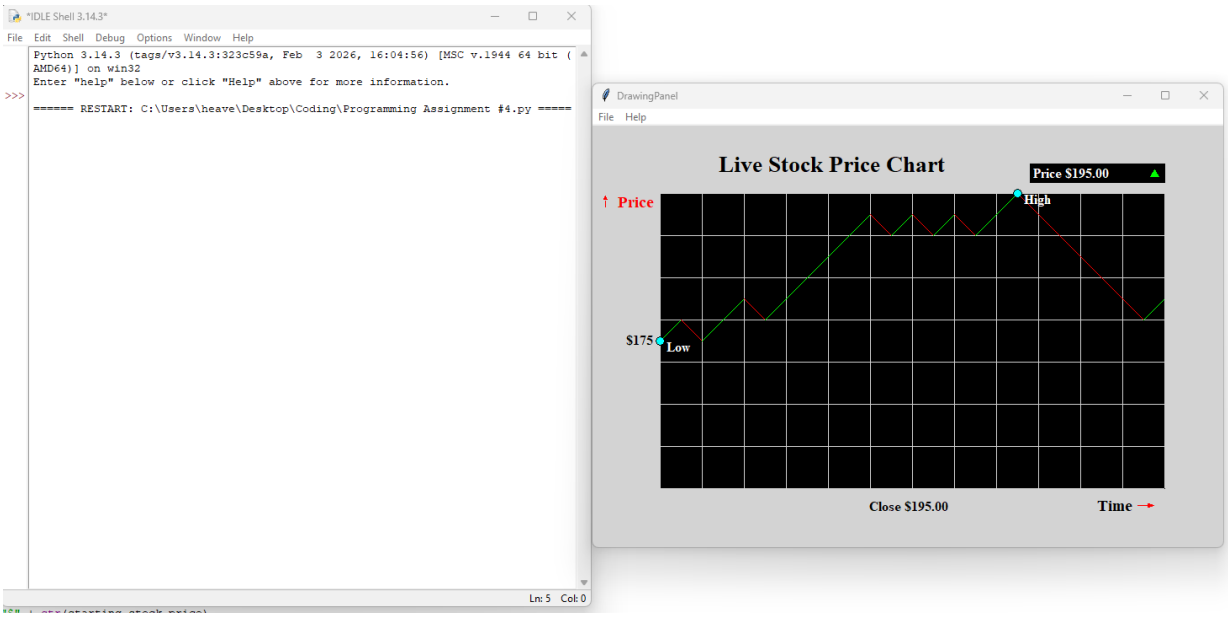
Programming Assignment #4.py - C:\Users\heave\Desktop\Coding\Programming Assignment #4.py (3.14.3)

File Edit Format Run Options Window Help

```
        direction = random.choice([-1, 1])
        next_y = y + direction * dy
        if next_y < top or next_y > bottom:
            direction = -direction
            next_y = y + direction * dy

        if direction < 0:
            price += price_delta
        else:
            price -= price_delta

        draw_price(panel, price, right)

        if direction < 0:
            panel.set_color("green")
        else:
            panel.set_color("red")
        panel.draw_line(x, y, x + dx, next_y)
        panel.sleep(100)

        x += dx
        y = next_y

        if y < highest_y:
            highest_x, highest_y = x, y
        if y > lowest_y:
            lowest_x, lowest_y = x, y
    return(highest_x, highest_y), (lowest_x, lowest_y), price

# Creates the box to the top right to show price as it animates the line
def draw_price(panel, price, right):
    dollars = int(price)
    text= "Price $" + str(dollars) + ".00"
    label_w = 160
    label_h = 22
    label_x = right - label_w
    label_y = 45

    panel.fill_rect(label_x, label_y, label_w, label_h, "black", outline="black")
    panel.set_color("white")
    panel.font= "Times 12 bold"
    panel.draw_string (text, label_x + 4, label_y + 2)

    triangle_x = label_x + label_w - 18
    triangle_y = label_y + label_h // 2
    if price > starting_stock_price:
        triangle_color = "green"
        y1 = triangle_y + 5
        y2 = triangle_y + 5
        y3 = triangle_y - 5
    else:
        triangle_color = "red"
        y1 = triangle_y - 5
        y2 = triangle_y - 5
        y3 = triangle_y + 5

    if price !=starting_stock_price:
        panel.fill_polygon(triangle_x, y1, triangle_x + 12, y2, triangle_x + 6, y3, fill=triangle_color)

def main():

    # Settings
    width = 750
    height = 500

    # Chart bounds
    left = 80
    top = 80
    right = width - 70
    bottom = height - 70
    panel = DrawingPanel(width, height, "light grey")

    # Title
    panel.set_color("black")
    panel.font = "Times 20 bold"
    panel.draw_string("Live Stock Price Chart", 150, 30)

    #Label Time Axis
    panel.font = "Times 14 bold"
    panel.draw_string("Time", right - 80, bottom + 10)

    arrow_x = right - 20
    arrow_y = bottom + 20
```

```
    panel.set_color("red")
    panel.fill_polygon(arrow_x, arrow_y - 3, arrow_x, arrow_y +3, arrow_x + 8, arrow_y, fill="red")
    panel.draw_line(arrow_x - 12, arrow_y, arrow_x, arrow_y)

    #Label Price Axis
    panel.font = "Times 14 bold"
    panel.draw_string("Price", left - 50, top)
    arrow_x = left - 65
    arrow_y = top + 8
    panel.set_color("red")
    panel.fill_polygon(arrow_x - 3, arrow_y, arrow_x + 3, arrow_y, arrow_x, arrow_y - 6, fill="red")
    panel.draw_line(arrow_x, arrow_y + 8, arrow_x, arrow_y)


    #Sets the initial stock price to $175
    starting_price_text = "$" + str(starting_stock_price)
    starting_y = (top + bottom) // 2 - 10
    panel.font = "Times 12 bold"
    panel.set_color("black")
    panel.draw_string(starting_price_text, left - 40, starting_y)


    # Draw grid
    draw_grid_frame(panel, left, top, right, bottom)

    #calls upon the functions to Simulates the "price of the stock over time", lowest price, highest price, and closing price
    high_point, low_point, final_price = animate_line(panel, left, top, right, bottom)
    draw_marker(panel, high_point[0], high_point[1], "High")
    draw_marker(panel, low_point[0], low_point[1], "Low")
    draw_close_label(panel, final_price, left, right, bottom)

main()
```

## Code Output:




**Explanation**: In this programming assignment we are tasked with creating a drawing panel drawing that simulates the price of a stock over a certain period of time. After the stock price has been simulated, it then highlights the highest price and lowest price the stock

was at as well as its price when the market closed. The program 24 animations using a while loop to simulate time and each time the program is run the stock price is randomly animated (positive or negative). The most challenging part that I found doing was animating the line to simulate the price of the stock over time. I honestly did not know where to really begin but I reviewed class notes and started from there and then when I got stuck, I used AI. AI had some wacky suggestions that I had no clue so I had to keep iterating upon it until it utilized a form of while loop for the animation which I started to understand. From where I worked through it slowly through trial and error until I was able to finally get the animation to work properly. Another thing that got me was because I think the AI was using variables that didn't really stick in my head so when I converted variables that made more sense to me, it became easier to process and manage the code.