Week 4 Demo Lecture Exercise Series

Example used in demo lecture

```
from graphix import Window, Rectangle, Circle, Point, Text
import time

def move_car():
    win = Window("Moving Car", 400, 400)

    sentence = "The car is now moving to the left"
    words = sentence.split()  # Splits sentence into words

    # Car body
    car_body = Rectangle(Point(100, 150), Point(300, 200))
    car_body.fill_colour = 'blue'
    car_body.draw(win)

    # Car top
    car_top = Rectangle(Point(140, 120), Point(260, 150))
    car_top.fill_colour = 'lightblue'
    car_top.draw(win)

    # Car wheels
    wheel1 = Circle(Point(130, 210), 20)
    wheel1.fill_colour = 'black'
    wheel1.draw(win)

    wheel2 = Circle(Point(270, 210), 20)
    wheel2.fill_colour = 'black'
    wheel2.draw(win)

    # Text box for displaying the sentence as it's revealed
    text_box = Text(Point(200, 100), "")  # Initial empty text
    text_box.draw(win)

    step = 5
    revealed_sentence = ""  # This will hold the revealed portion of the
sentence
    # Move the car to the right
    for word in words:
```

```
        win.get_mouse()
        car_body.move(step, 0)  # Move car body
        car_top.move(step, 0)   # Move car top
        wheel1.move(step, 0)    # Move left wheel
        wheel2.move(step, 0)    # Move right wheel
        # Add the next word to the revealed sentence
        revealed_sentence += word + " "
        text_box.text = revealed_sentence

    win.get_mouse()  # Wait for user to click before closing
    win.close()

move_car()
```

---

1. Drawing a Rectangle (Car Body)

Objective: Draw a rectangle representing the body of a car.

```
from graphix import Window, Rectangle, Point

def draw_rectangle():
    win = Window("Car Body", 400, 400)
    car_body = Rectangle(Point(100, 150), Point(300, 200))  # Car body between
two points
    car_body.fill_colour = 'blue'  # Set the fill color
    car_body.draw(win)  # Draw the car body
    win.get_mouse()  # Wait for a mouse click
    win.close()  # Close the window

draw_rectangle()
```

Concepts: Rectangle creation, setting fill color, drawing shapes.
Variations: Allow the user to input car body dimensions and colors.

---

2. Drawing a Single Circle (Wheel)

Objective: Draw one circle representing a wheel of the car.

```
from graphix import Window, Circle, Point
def draw_wheel():
    win = Window("Car Wheel", 400, 400)
    wheel = Circle(Point(150, 210), 20)  # Single wheel
    wheel.fill_colour = 'black'
    wheel.draw(win)

    win.get_mouse()  # Wait for a mouse click
    win.close()  # Close the window

draw_wheel()
```

Concepts: Circle creation, setting fill color, positioning the wheel.
Variations: Allow the user to set the wheel position and size interactively.

---

3. Drawing a Car (Combination of Rectangle and Circles)

Objective: Combine a rectangle and two circles to form a car.

```
from graphix import Window, Rectangle, Circle, Point

def draw_car():
    win = Window("Car Drawing", 400, 400)

    # Draw the car body (rectangle)
    car_body = Rectangle(Point(100, 150), Point(300, 200))
    car_body.fill_colour = 'blue'
    car_body.draw(win)

    # Draw the top part of the car (smaller rectangle above the body)
    car_top = Rectangle(Point(140, 120), Point(260, 150))
    car_top.fill_colour = 'lightblue'
    car_top.draw(win)
```

```
    # Draw the wheels (circles)
    wheel1 = Circle(Point(130, 210), 20)  # Left wheel
    wheel1.fill_colour = 'black'
    wheel1.draw(win)

    wheel2 = Circle(Point(270, 210), 20)  # Right wheel
    wheel2.fill_colour = 'black'
    wheel2.draw(win)

    win.get_mouse()  # Wait for a mouse click
    win.close()  # Close the window

draw_car()
```

Concepts: Combining multiple shapes to create a more complex object. **Variations:** Allow the user to change car colors or customize positions interactively.

---

4. Moving the Car

Objective: Move the car to the right with smooth animation.

```
from graphix import Window, Rectangle, Circle, Point
import time

def move_car():
    win = Window("Moving Car", 400, 400)

    # Car body
    car_body = Rectangle(Point(100, 150), Point(300, 200))
    car_body.fill_colour = 'blue'
    car_body.draw(win)

    # Car top
    car_top = Rectangle(Point(140, 120), Point(260, 150))
    car_top.fill_colour = 'lightblue'
    car_top.draw(win)

    # Car wheels
    wheel1 = Circle(Point(130, 210), 20)
    wheel1.fill_colour = 'black'
    wheel1.draw(win)
```

```
    wheel2 = Circle(Point(270, 210), 20)
    wheel2.fill_colour = 'black'
    wheel2.draw(win)

    # Move the car to the right
    for _ in range(100):  # Move the car 100 pixels to the right
        time.sleep(0.05)  # Delay to create animation effect
        car_body.move(1, 0)  # Move car body
        car_top.move(1, 0)   # Move car top
        wheel1.move(1, 0)    # Move left wheel
        wheel2.move(1, 0)    # Move right wheel

    win.get_mouse()  # Wait for user to click before closing
    win.close()

move_car()
```

Concepts: Animating objects, using loops, moving multiple shapes together. **Variations**: Change the speed or direction of movement, allow user input to control movement.

---

5. Combining All Car Parts in a List

Objective: Combine all parts of the car (body, top, and wheels) into a list and move them using a loop.

```
from graphix import Window, Rectangle, Circle, Point
import time

def move_car():
    win = Window("Moving Car with List", 400, 400)

    # Create car parts
    car_body = Rectangle(Point(100, 150), Point(300, 200))
    car_body.fill_colour = 'blue'

    car_top = Rectangle(Point(140, 120), Point(260, 150))
    car_top.fill_colour = 'lightblue'
```

```
    wheel1 = Circle(Point(130, 210), 20)  # Left wheel
    wheel1.fill_colour = 'black'

    wheel2 = Circle(Point(270, 210), 20)  # Right wheel
    wheel2.fill_colour = 'black'

    # Draw all parts
    car_body.draw(win)
    car_top.draw(win)
    wheel1.draw(win)
    wheel2.draw(win)

    # Put all parts into a list
    car_objects = [car_body, car_top, wheel1, wheel2]

    # Move all parts of the car
    for _ in range(100):  # Move the car 100 pixels to the right
        time.sleep(0.05)  # Delay for animation effect
        for part in car_objects:
            part.move(1, 0)  # Move each part to the right by 1 pixel

    win.get_mouse()  # Wait for user to click before closing
    win.close()

move_car()
```

Concepts:

- Grouping objects into a list.
- Looping through the list to manipulate (move) multiple objects at once.
- Animating objects together.

Variations:

- Allow user control over movement direction (left, right, up, down).
- Experiment with different speeds or distances by adjusting the `range()` and `time.sleep()` values.