

M8 Classwork Mon

- **Introduction to classes** (Go through [vector.py](https://jhu.instructure.com/courses/105759/files/14855150/preview) (<https://jhu.instructure.com/courses/105759/files/14855150/preview>))
- **Part A: Object-Oriented Code Tracing**

Assume we have two classes named Point and Circle. Class Point has two instance variables of type float: x and y which represent x and y coordinates. Circle class has a center of type Point (i.e., x and y coordinates of the circle) and an instance variable named radius of type float. What is the following piece of code trying to decide? What messages would be sensible to print?

```
import math
p1 = Point(-2, 2)
c1 = Circle(p1, 4)
p2 = Point(4, 2)
c2 = Circle(p2, 8)

d = (c2.getCenter().getX() - c1.getCenter().getX())**2 + (c2.getCenter().getY() - c1.getCenter().getY())**2

if math.sqrt(d) >= c1.getRadius() + c2.getRadius():
    # print a sensible message here

elif math.sqrt(d) < c1.getRadius() + c2.getRadius():
    # print a sensible message here
```

- **Part B: Object-Oriented Implementation** (**Paired programming Activity - see below**)
 - **Task 1** (**First Half - Navigator: Student A, Driver: Student B**):
 - **Define the `Point` class:**
 - Write the constructor with appropriate parameters.
 - Implement the **getter (accessor)** and **setter (mutator)** methods for attributes.
 - Ensure correct use of the `self` keyword.
 - **Define the `Circle` class:**
 - Write the constructor with relevant parameters.
 - Implement the **getter and setter** methods for `radius` and `center` attributes.
 - **Task 2** (**Second Half - Navigator: Student B, Driver: Student A**):

- Implement the `getArea()` method in the `Circle` class, which calculates and returns the area using the formula πr^2 .
- Implement the `getPerimeter()` method in the `Circle` class, which calculates and returns the perimeter using the formula $2\pi r$.
- Ensure everything is working as expected with proper use of the `self` keyword.

- **Part C:**

- **Task 3 (Navigator: Student A, Driver: Student B):**

- Write the method `contains(otherCircle)` in the `Circle` class that returns `True` if the other circle is enclosed by this circle. Otherwise it should return `False`

- **Part D:**

- **Task 4 (Navigator: Student B, Driver: Student A):**

- Implement the `if __name__ == "__main__"` block to test the code. This should involve creating some `Point` and `Circle` objects and calling the methods (`getArea()`, `getPerimeter()`, `contains()`) to verify the correctness of the implementation.

- **Paired Programing two roles function:**

1. Driver

- **Role:** The driver is responsible for **writing the code**. They are the ones physically typing and implementing the solution.
- **Focus:** The driver's attention is on the immediate problem at hand, ensuring the syntax, logic, and structure of the code are correct.
- **Action:** The driver implements the solution as discussed with the navigator and might ask clarifying questions or discuss possible solutions.
- **Learning Benefit:** The driver practices translating ideas into working code, improving their problem-solving and programming skills.

2. Navigator

- **Role:** The navigator is responsible for **guiding** the driver. They actively observe the code being written and think ahead about potential issues or improvements.
- **Focus:** The navigator focuses on the **bigger picture**, making sure the driver is following the overall plan or structure of the assignment.
- **Action:** The navigator checks for logic errors, provides suggestions for improvement, and ensures that best practices are followed. They offer guidance on how to proceed, while the driver executes.

- **Learning Benefit:** The navigator develops skills in code review, critical thinking, and overall problem-solving. They learn how to strategize and identify potential problems without being bogged down in syntax.
-