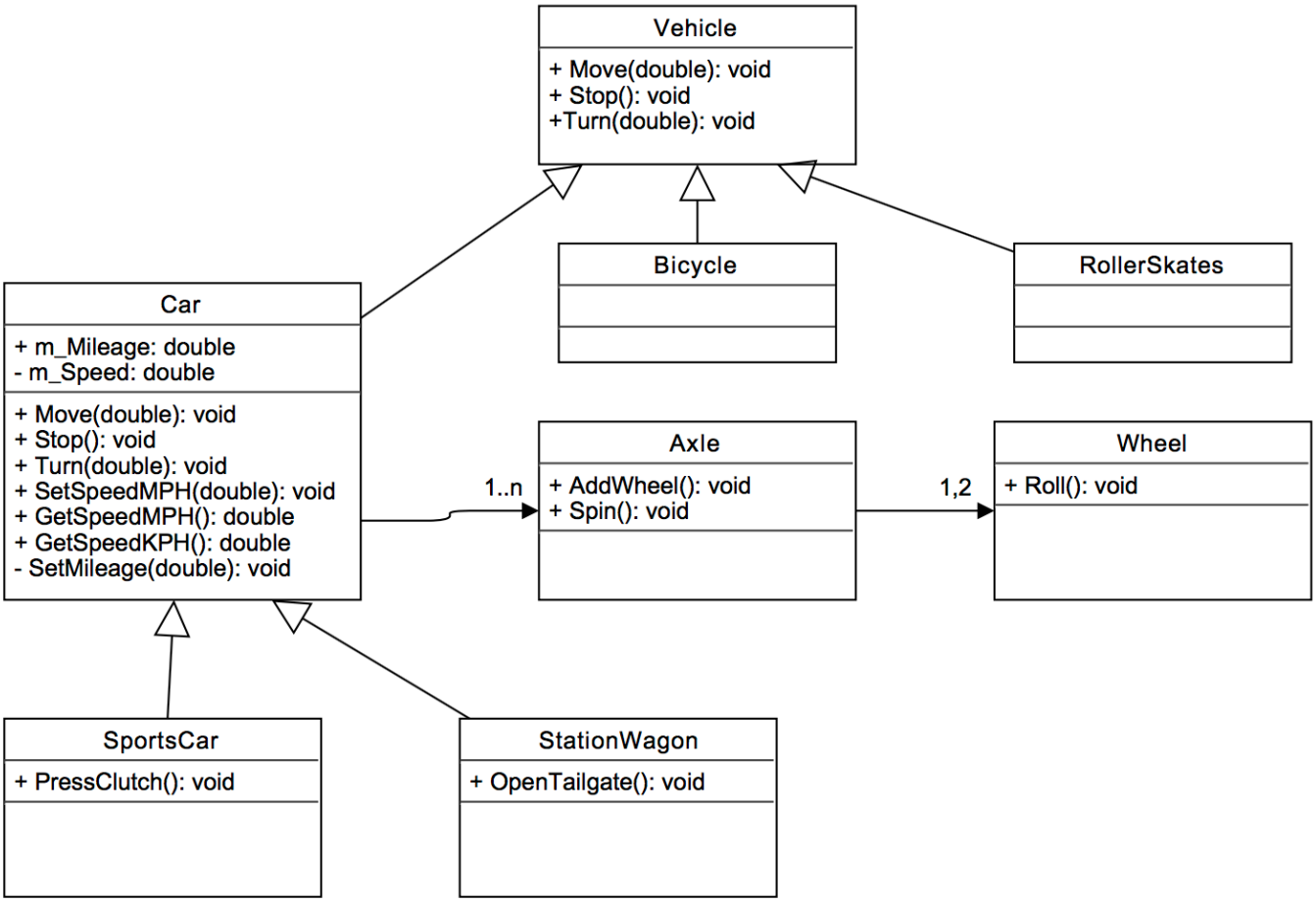
COSC 412

Fall 2024

Assignment 2

Due 5th November 2024 11:59 PM

Question 1 (10 points)



a) ***(4 points)*** Given the above class diagram, list all the methods that the Sports car class has.

b) ***(2 points)*** The station Wagon class has the "SetMileage" method. (True/False). Justify your answer.

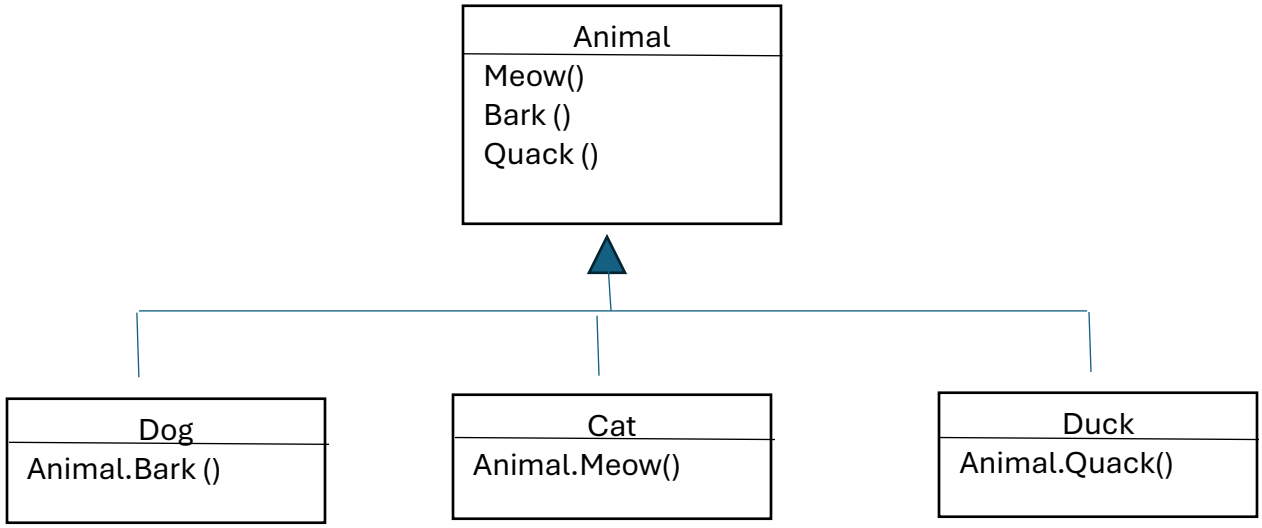
c) ***(4 points)*** What does generalization explain with respect to class diagrams? List all the generalization relationships in the given class diagram.

Question 2 (10 points)

Take a look at the following class diagram and answer the following questions.

a) ***(5 points)*** What does Open-Close principle suggest? Apply Open-Close principle to the above problem and give us the new class diagram.

b) ***(5 points)*** What does the Dependency Inversion principle suggest? Apply the Dependency Inversion principle to the above problem and give us the new class diagram.



Question 3 (10 points)

We have a "Rectangle" interface and a "Square class". Use your understanding of the Adapter Pattern to use square class's object as a rectangle.

```
class Square {
    private int side;

    public void setSide(int side) {
        this.side = side;
    }

    public int getSide() {
        return side;
    }
}
```

```
interface Rectangle {
    void setWidth(int width);
    void setHeight(int height);
    int getWidth();
    int getHeight();
}
```

a) ***(4 points)*** Design a Square adapter

b) ***(4 points)*** Given that we cannot alter the performance, what would be the output of the below main method. Also, identify how this square adapter is failing to perform like a rectangle

```
public class Performance {
    public static void clientCode(Rectangle rectangle) {
        int width = 5;
        int height = 10;

        rectangle.setWidth(width);
        rectangle.setHeight(height);

        System.out.println("Width: " + rectangle.getWidth());
        System.out.println("Height: " + rectangle.getHeight());
        System.out.println("Area: " + rectangle.getWidth() * rectangle.getHeight());
    }

    public static void main(String[] args) {
        Square square = new Square();
        SquareAdapter adapter = new SquareAdapter(square);

        System.out.println("Client code is working with the Square via the Rectangle
interface:");
        clientCode(adapter);
    }
}
```

c) ***(2 points)*** Give the class diagram for this adapter pattern